INFORMÁTICA BÁSICA

Ed-mundo

Realização

The Human Project

Apoio

SECRETARIA DE ESTADO DO DESENVOLVIMENTO ECONÔMICO E DA CIÊNCIA E TECNOLOGIA

SERGIPE GOVERNO DO ESTADO

Parceria

ATLAS RENEWABLE ENERGY

SUMÁRIO

MÓDULO 1

INFORMÁTICA BÁSICA

Primeiros Passos

Neste módulo, aprenderemos conceitos fundamentais sobre computadores e o mundo digital. Abordaremos temas como hardware, sistemas operacionais, redes, softwares, navegadores, diretórios, metaverso e muito mais. Além disso, realizaremos atividades práticas e desafios projetados para estimular a curiosidade e a criatividade dos alunos.

No final do módulo, faremos uma revisão em forma de quiz para consolidar o aprendizado. Espera-se que, ao término deste módulo, os alunos compreendam os conceitos básicos apresentados e tenham desenvolvido habilidades motoras essenciais para o uso do computador, facilitando a progressão nos módulos seguintes.

AULA 01

“Quem é você?” + Autorretrato

Materiais necessários:

Papel, lápis, caneta, borracha, lápis para colorir, projetor.

Duração:

1 hora e 30 minutos

Objetivo:

Proporcionar aos participantes uma reflexão lúdica sobre os motivos de sua participação no projeto, seus gostos pessoais, habilidades e possibilidades de atuação. Além disso, promover a compreensão da importância do autoconhecimento e da autoestima, permitindo que reconheçam e valorizem sua própria identidade por meio da criação de um autorretrato.

ENGAJAR

É importante iniciar a aula se apresentando e conhecendo melhor seus novos alunos.

DICA: Você pode utilizar para esse momento as dinâmicas presentes no baralho de inteligência emocional e realizar atividades para quebrar o gelo.

INVESTIGAR

Após a apresentação inicial, apresente aos alunos o que eles vão aprender no curso e explique um pouco sobre os temas.

TEMAS: Informática Básica, Figma, Lógica de Programação, HTML/CSS, JavaScript. FlutterFlow, Banco de Dados, Robótica.

Enquanto fala sobre os temas, interaja com os alunos, perguntando se algum deles já teve contato com o tópico. Mostre exemplos práticos utilizando recursos como sites, imagens e vídeos.

AGIR

Aplique duas atividades com os alunos nesse momento.

ATIVIDADE 1

QUEM É VOCÊ?

1. Explique a importância do autoconhecimento para a criação de uma ideia de negócio, evidenciando que isso pode ser divertido e uma possibilidade de conhecer mais sobre os participantes do projeto;

2. Cada participante deverá receber uma folha para responder as quatro questões sobre si de forma anônima;

   1ª Pergunta: Por que estou participando do Ed-Mundo ?

   2ª Pergunta: Quais são as coisas que mais gosto de fazer?

   3ª Pergunta: O que as pessoas me pagariam para fazer?

   4ª Pergunta: O que as pessoas dizem que eu faço muito bem?

3. Após responder as quatro questões, cada participante deverá colocar seu papel dentro de uma caixa ou sacola;

4. O(a) reaplicador(a) em uma cartolina deverá desenhar uma tabela com 5 colunas e a quantidade de linhas de acordo com o número de participantes da atividade;

5. Além de inserir as quatro perguntas já respondidas pelos participantes, o(a) reaplicador(a) deverá escrever na quinta coluna a pergunta "Considerando as quatro perguntas anteriores, qual negócio essa pessoa poderia criar?";

6. O(a) reaplicador(a) deverá retirar cada papel da sacola ou caixa, e preencher a tabela conforme as respostas. Após ler as respostas das quatro perguntas, o(a) reaplicador(a) deverá perguntar para os participantes, com base na análise do que foi respondido, qual seria a resposta ideal para a nova pergunta da quinta coluna. E assim, deverá ser feito com todos os papéis preenchidos;

7. Para finalizar, pode ser questionado ao grupo se os participantes conseguem identificar de quem são as respostas, possibilitando um clima de descontração e de curiosidade.

ATIVIDADE 1

AUTORRETRATO

1. Perguntar aos participantes o que eles entendem por autorretrato e elaborar uma resposta a partir dos comentários deles;

2. Pedir para que façam seu autorretrato, buscando destacar suas características mais marcantes, mas sem esquecer dos detalhes;

3. Solicitar que a partir do desenho, escrevam 3 características próprias que gostam muito (positivas);

4. Perguntar o que os participantes acharam da experiência e explicar que essa atividade é importante para aproximação, aceitação e reconhecimento das próprias características.

AULA 02

diretórios/hardware/software e navegador

Materiais necessários:
Notebook, internet, computador, projetor.

Duração:
1 hora e 30 minutos

Objetivo:
Fornecer o primeiro contato com um computador apresentando aos alunos conceitos básicos.

ENGAJAMENTO

OBS: Antes de iniciar a aula, apresente aos alunos os cuidados necessários que devemos ter ao manusear os notebooks e computadores:

Ventilação adequada:
Mantenha as entradas de ar do notebook desobstruídas para evitar o superaquecimento. Use o notebook em superfícies planas para permitir a circulação de ar adequada.

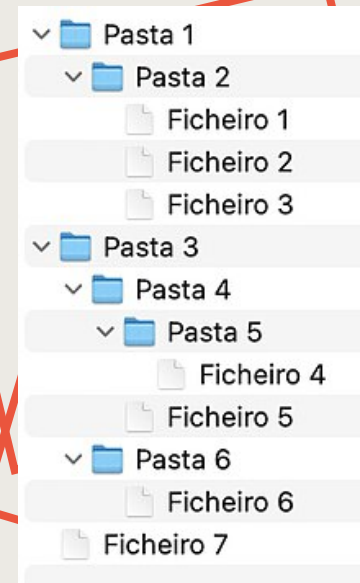
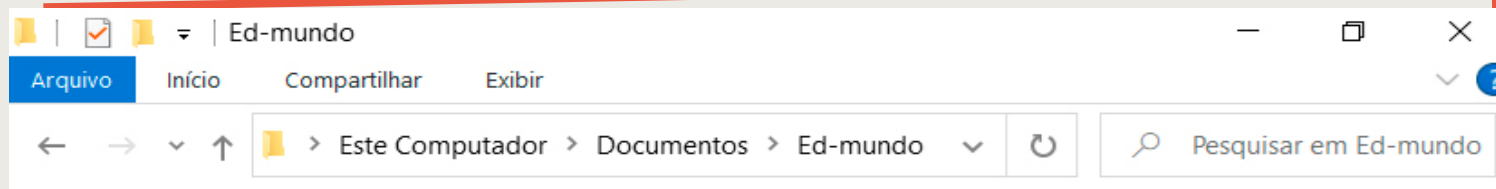
Evite líquidos:
Mantenha líquidos longe do notebook. Mesmo pequenos derramamentos podem causar danos graves aos componentes internos.

Uso da fonte de alimentação:
Utilize a fonte de energia fornecida pelo fabricante ou uma compatível. O uso de fontes inadequadas pode danificar o notebook.

Desligamento adequado:
Desligue completamente o notebook antes de transportá-lo ou guardá-lo. Isso evita perda de dados e problemas de inicialização.

Local de armazenamento:
Evite locais extremamente quentes, frios ou úmidos para armazenar o notebook. Mantenha-o em um ambiente seco e seguro quando não estiver em uso.

AUTORRETRATO

Diretórios, também conhecido como pastas, no contexto da computação e tecnologia da informação, referem-se a estruturas de organização e armazenamento de arquivos em sistemas de computador. Eles desempenham um papel fundamental na organização, na hierarquia dos dados em um sistema operacional e pode conter arquivos individuais e, até mesmo, outros diretórios, formando uma estrutura hierárquica ou em árvore. Isso permite uma organização lógica e eficiente dos dados, facilitando a localização e o acesso aos arquivos.

Fonte:
https://pt.wikipedia.org/wiki/Diret%C3%B3rio_%28computa%C3%A7%C3%A3o%29

HARDWARE

É um termo técnico que se refere à parte física de computadores e outros sistemas microeletrônicos. É tudo o que a gente consegue tocar.
Ex.: Teclado, mouse, monitor, impressora, placas, pen drive, webcam, roteador. Porque existem no mundo real.

Fonte:
https://ipti.gitbook.io/

SOFTWARE

Um conjunto de instruções que permite controlar um aparelho eletrônico. É a parte lógica, tudo aquilo que a gente não consegue colocar as mãos. Não dá para pegar o navegador ou um site.

Obs.: Para ter acesso ao software, é preciso ter hardware.

Ex.: No celular, por mais que a gente toque na tela, o toque é no telefone e não no site. O sensor é a tela.

FIRMWARE

A função do firmware é armazenar todas as informações para que o equipamento funcione corretamente. O firmware é armazenado permanentemente num circuito integrado (chip) de memória de hardware.
É um tipo de programa, mas são programas que ficam nas placas para controlar os componentes, mas diferente de um pen drive, esses só têm memória suficiente para guardar esses programas.

NAVEGADOR

Navegadores de internet são aplicativos de software projetados para acessar e exibir informações. Eles são a principal interface entre os usuários e todo o conteúdo disponível na internet.

Fonte:
https://setupgsm.com.br/2017/04/12/microsoft-edge-bate-chrome-e-mozila-firefox-em-teste-de-bateria/

O Google, é o principal mecanismo de busca na Internet, oferece diversas funcionalidades que podem ser aproveitadas para refinar e otimizar suas buscas.

1. Pesquisas básicas:

Palavras-chave: utilize palavras-chave relevantes para descrever o que você procura. Por exemplo: tecnologia quântica explicada.

Imagens: use a guia “Imagens” para pesquisar visualmente. É possível aprimorar sua busca utilizando ferramentas, podendo filtrar a busca por tamanho da imagem, cor, tipo, tempo e direitos de uso.

Vídeos: explore a guia “Vídeos” para encontrar conteúdo multimídia.

Notícias: refine suas pesquisas por notícias atuais usando a guia “Notícias”.

Shopping: utilize a guia “Shopping” para pesquisa de preço de mercado.

2. Pesquisas avançadas:

Pesquisa exata: Coloque uma frase entre aspas para obter resultados exatos. Por exemplo: “inteligência artificial”.

Pesquisa por site: Limite suas pesquisas a um site específico usando “site:” seguido do domínio. Por exemplo: site:nytimes.com mudanças climáticas.

Pesquisa coringa: usa-se o * em palavras que você não recorda. Por exemplo: Tobias Ro*.

Pesquisa de exclusão: exclui da pesquisa palavras indesejadas. Por exemplo: Tobias –Rodrigues.

Pesquisa de inclusão: inclui uma palavra no título da pesquisa. Por exemplo: Tobias +Rodrigues.

Pesquisa em títulos: adiciona allintitle: antes do termo em questão. Por exemplo: allintitle:tobias.

Pesquisa de formato: adiciona filetype: depois do termo em questão. Por exemplo: informática filetype:pdf.

Definições: Descubra definições usando “define:” antes da palavra. Por exemplo, “define: inteligência artificial”.

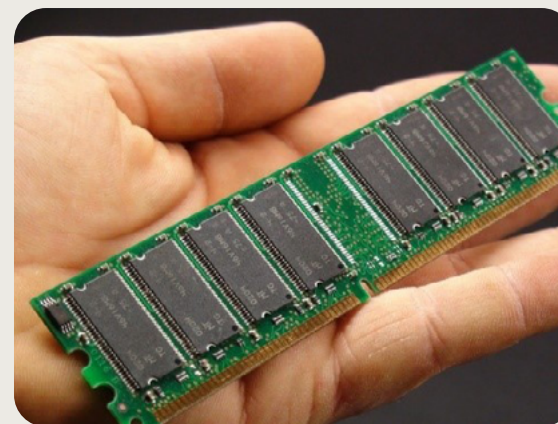
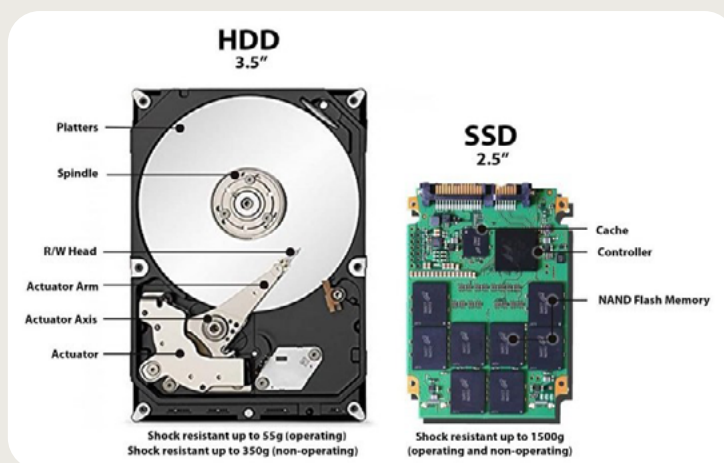
Calculadora e conversões: Utilize o Google para cálculos matemáticos ou conversões. Por exemplo, “25 * 4” ou “10 dólares em euros”.

INVESTIGAR

Se houver um computador desmontado disponível em sala, distribua alguns dos hardwares para os alunos (que devem estar separados em grupos) e peça para que eles façam pesquisas sobre a função daquele componente e, em seguida, para que apresentem os resultados de suas buscas.
Caso não tenha um computador desmontado, peça para que eles pesquisem exemplos de hardwares no navegador.

Proposta de solução

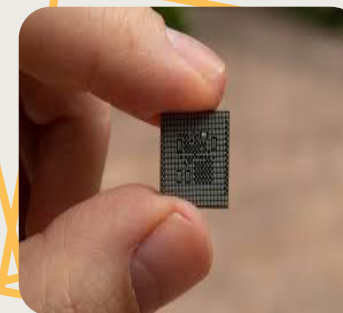
Memória RAM: A memória RAM é um tipo de memória de acesso rápido e temporário usada em computadores e dispositivos eletrônicos. Ela armazena dados e programas em uso enquanto o dispositivo está ligado, permitindo acesso rápido e essencial para o funcionamento eficiente do sistema.

Fonte:
https://ipti.gitbook.io/

Fonte:
https://ipti.gitbook.io/

HD e SSD: HDs são armazenamentos magnéticos mais econômicos e de alta capacidade, mas lentos. Já os SSDs usam chips de memória, são mais rápidos e duráveis, porém tendem a ter capacidades menores e custo mais elevado.

Processador: O processador é o cérebro do computador. Ele transforma informações em uma linguagem que o computador entende (0101). Tem o formato de um chip e transforma dados em informações, por exemplo, carrega páginas, faz download, abre arquivos e executa programas. A velocidade que seu computador abre ou executa programas é em parte responsabilidade do processador.

Fonte:
https://ipti.gitbook.io/

Fonte:
https://ipti.gitbook.io/

AGIR 1

Desafio:

Pesquisar e comparar diferentes navegadores de internet para compreender suas características, funcionalidades e diferenças.

Material necessário:

Notebook;
Internet.

Proposta de solução:

Google Chrome: Muito popular, rápido e tem muitos complementos. É fácil de usar e funciona bem.

Mozilla Firefox: Prioriza a privacidade e a segurança, tem muitos extras e é feito por muitos desenvolvedores.

Microsoft Edge: Rápido, compatível com Windows, possui recursos como anotações e modo de leitura.

Safari: Funciona bem nos dispositivos da Apple, economiza energia e se conecta facilmente entre dispositivos.

Opera: Tem recursos inovadores como VPN, bloqueio de anúncios e economia de bateria. É personalizável e seguro.

Brave: Protege a privacidade bloqueando anúncios e rastreadores automaticamente. É rápido e baseado no código aberto.

Vivaldi: Muito personalizável, permite ajustes em todos os aspectos, é flexível e ótimo para produtividade.

AGIR 2

Desafio de criatividade:

Desafie os alunos a criarem um “Navegador do Futuro”. Eles podem imaginar e desenhar como seria um navegador inovador, incluindo funcionalidades que tornariam a navegação mais fácil, segura e divertida. Podem pensar em design, recursos e até mesmo criar um nome para o navegador.

Material necessário:

- Papel;
- Caneta ou lápis;
- Borracha;
- Lápis de cor.

AULA 03

Sistemas Operacionais

Tema:
Tendências, tecnologias e sistemas operacionais.

Materiais necessários:
Notebook, internet, projetor.

Duração:
1 hora e 30 minutos

Objetivo:
Contextualizar os alunos sobre o avanço da tecnologia e apresentar ferramentas de pesquisa.

ENGAJAR

As constantes mudanças e avanços tecnológicos estão influenciando profundamente nossa sociedade. Algumas tendências emergentes incluem a inteligência artificial, a Internet das Coisas e o desenvolvimento do metaverso.

A inteligência artificial é um ramo da computação que visa criar sistemas capazes de pensar e agir de forma semelhante aos seres humanos. Ela abrange diferentes áreas, como o Aprendizado de Máquina e Aprendizado Profundo, permitindo que máquinas aprendam, resolvam problemas e tomem decisões. Ela é aplicada em assistentes virtuais, diagnósticos médicos e carros autônomos. No entanto, há preocupações éticas, como privacidade e tendências enviesadas nos algoritmos.

A Internet das Coisas (IoT) é a integração de dispositivos do mundo real à internet, possibilitando a coleta e troca de dados entre eles. Esses dispositivos podem ser desde eletrodomésticos até sistemas industriais, usando sensores para coletar informações e interagir entre si. Essa tecnologia oferece vantagens em setores como saúde, automação residencial e agricultura, porém suscita questões de segurança e privacidade devido à grande quantidade de dados transmitidos e armazenados.

Sistemas Operacionais (SO) são softwares que fornecem uma interface entre o hardware do computador e o usuário. Eles são essenciais para gerenciar recursos, executar programas e facilitar a interação com o computador.

Os sistemas operacionais são essenciais para diversas tarefas:

Gestão de recursos: Controlam o uso eficiente da CPU, memória, armazenamento e periféricos, garantindo que os programas obtenham o recurso necessário.

Interface com o usuário: Oferecem uma forma de interação com o computador, seja por uma interface visual ou por comandos de texto.

Execução de aplicativos: Permitem a inicialização, execução e encerramento de softwares, gerenciando seu funcionamento.

Organização de dados: Estruturam e administram como os dados são armazenados, acessados e organizados nos dispositivos.

Há vários sistemas operacionais conhecidos, como Windows, macOS, Linux, iOS e Android, cada um com suas particularidades e usos específicos. Eles atendem desde computadores pessoais e laptops até dispositivos móveis, servidores e sistemas integrados.

Fonte:
https://www.buscape.com.br/notebook/conteudo/o-que-e-sistema-operacional

INVESTIGAR

O metaverso, que tem ganhado destaque recentemente, representa uma dimensão digital, uma realidade virtual compartilhada e expansiva que vai além das experiências convencionais online. Neste universo virtual, os limites da realidade são desafiados, proporcionando interações, comunidades e possibilidades únicas.

Atividade: O aluno deverá pesquisar mais a fundo o universo do Metaverso. Investigando as aplicações práticas, os setores que estão sendo transformados, as tecnologias que impulsionam essa evolução e as implicações éticas e sociais dessa nova fronteira digital. Além disso, pense em como o Metaverso pode impactar áreas como entretenimento, educação, negócios e interações sociais. Como esse conceito pode moldar o futuro da interação humana? Ao final, peça para cada aluno falar um pouco sobre suas impressões da pesquisa e seu ponto de vista sobre os temas abordados.

AGIR 1

Desafio:

Objetivo: Permitir que os adolescentes investiguem diferentes sistemas operacionais e entendam suas características principais.

Divida a turma em grupos e atribua a cada grupo um sistema operacional comum (como Windows, MacOS, Linux, iOS ou Android). Cada grupo deverá listar as principais características e compartilhar suas descobertas com os demais.

SAIBA MAIS

Existem inúmeros empreendedores extraordinários no campo da tecnologia, cada um trazendo consigo uma visão singular e contribuições únicas. Você conhece alguns deles?

Bill Gates é um dos grandes visionários da informática. Fundou a Microsoft em 1975 com Paul Allen e teve um papel crucial em tornar os computadores pessoais populares. Foi um dos primeiros a ver como os computadores podiam ser úteis para todos. Criou sistemas operacionais como o MS-DOS e o Windows, mudando completamente nossa forma de usar computadores.

Steve Jobs, um dos maiores inovadores da tecnologia, co-fundou a Apple em 1976. Ele trouxe ao mundo produtos icônicos como o Macintosh, iPod, iPhone e iPad, revolucionando a informática pessoal. Jobs era obcecado pela perfeição no design e na união perfeita entre hardware e software. Sua abordagem revolucionária mudou completamente nossa relação com dispositivos eletrônicos, priorizando a experiência do usuário como nunca.

Mark Zuckerberg ficou conhecido como co-fundador do Facebook, uma das maiores redes sociais do mundo. Em 2004, enquanto estudava na Universidade Harvard, ele desenvolveu a plataforma para conectar alunos da universidade. O sucesso inicial levou a um crescimento acelerado, expandindo para outros campi e, posteriormente, para o público em geral.

AULA 04

Tema:
Ferramentas Google + Árvore de pensamento

Materiais necessários:
Papel, lápis, caneta, Notebook, internet, borracha e projetor.

Duração:
1 hora e 30 minutos

Objetivo:
Apresentar aos alunos as principais ferramentas do Google e aplicar uma atividade prática para exercitar a digitação, destacando funções como o uso da tecla Caps Lock e o atalho para interrogação no teclado. Promover, também, a reflexão sobre perspectivas de presente e futuro, incentivando a organização de pensamentos e interesses individuais.

ENGAJAR

Inicie a aula falando um pouco sobre as ferramentas do Google.

Texto referência:

As ferramentas do Google, como o Google Forms, Google Docs e outras, compõem um conjunto poderoso de aplicativos baseados na nuvem que oferecem uma ampla gama de funcionalidades para colaboração, criação de documentos, organização de informações e muito mais.

Vamos explorar algumas dessas funcionalidades:

GOOGLE FORMS

É uma ferramenta para criar formulários e pesquisas personalizadas de maneira simples e eficiente. Com ele, é possível criar questionários, enquetes, formulários de inscrição e coletar respostas de maneira organizada. Os resultados podem ser facilmente visualizados e analisados.

GOOGLE DOCS

É um processador de texto online que permite criar e editar documentos de texto. Ele oferece recursos semelhantes aos tradicionais programas de processamento de texto, como formatação de texto, inserção de imagens, tabelas, colaboração em tempo real e armazenamento na nuvem.

GOOGLE SHEETS

É uma planilha eletrônica que permite criar, editar e compartilhar planilhas online. Oferece recursos avançados para análise de dados, fórmulas, gráficos e integração com outras ferramentas do Google.

GOOGLE SLIDES

É uma ferramenta para criar apresentações de slides online, semelhante ao Microsoft PowerPoint. Permite criar slides, adicionar elementos visuais, como texto, imagens, gráficos, efeitos de transição e colaborar com outras pessoas em tempo real.

GOOGLE DRIVE

É o serviço de armazenamento na nuvem do Google, onde todos os arquivos criados nas outras ferramentas do Google são armazenados e organizados. Oferece espaço gratuito para armazenamento e permite o acesso aos arquivos em qualquer dispositivo conectado à internet.
Essas ferramentas são altamente colaborativas, possibilitando a edição e o trabalho simultâneo de várias pessoas nos documentos. Isso facilita a comunicação e o compartilhamento de informações entre equipes, escolas e grupos de trabalho. Além disso, por estarem baseadas na nuvem, proporcionam acesso conveniente a partir de qualquer dispositivo conectado à internet, dispensando a instalação de programas adicionais.

INVESTIGAR

Explore as ferramentas google durante a explicação do conteúdo, mostrando o que cada ferramenta é capaz de fazer como currículo, agendas, apresentações, álbum de fotos...

Nesse momento também, crie um e-mail para a turma junto com os alunos. O e-mail será utilizado de forma coletiva pelos alunos até o final do curso.

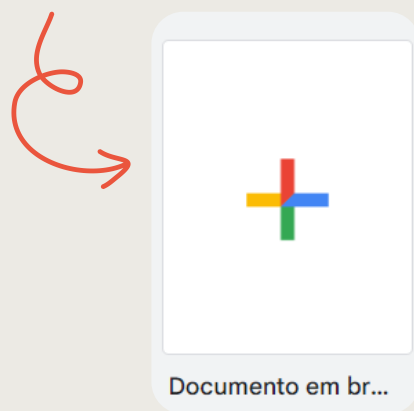
AGIR

Desafio:

Objetivo: Inicie a atividade explicando que temos pensamentos que podem ser negativos e que isso pode atrapalhar os participantes ao longo da caminhada no processo, sendo interessante transformar tais questionamentos em pensamentos positivos ou em oportunidades de melhorias. Por exemplo, substituir o "isso pode dar errado" por "o que fazer para dar certo?".

Usando as ferramentas google que aprendemos nessa aula, peça para que os alunos realizem a seguinte atividade:

1- Acessem o Google Docs e crie um arquivo em branco:

Documento em br...

2 - Peça para que os participantes escrevam a palavra "positivo".

Formatação:

Positivo: Fonte Arial, 12, cor verde, em Capslock e em negrito.

POSITIVO

Negativo: Fonte Arial, 12, cor Vermelho, em Capslock e em negrito.

NEGATIVO

3 – Agora peça para que os participantes reflitam e escrevam no documento criado sobre três tipos de pensamentos: positivos, negativos.

Obs.: Nesse momento os alunos podem explorar as fontes e funcionalidades das ferramentas.

4 – Após o preenchimento das perguntas, peça para que os participantes digitem as seguintes perguntas no documento:

"Qual é o seu propósito?", "Como você se descreve?", "Quais qualidades destacariam em você?" e "Em quais pontos você gostaria de melhorar?".

Formatação: Fonte Arial, 12, cor preto e em negrito.

Como você se descreve?

Qual o seu propósito?

Quais qualidades destacaria em você?

Em quais pontos você gostaria de melhorar?

5. Após o preenchimento de todas as questões apresentadas na atividade, o(a) reaplicador(a) deve indicar que os participantes fechem os olhos e reflitam sobre os seus objetivos pessoais e que pensem no complemento para a frase "Eu queria...";

6. Em seguida, recomenda-se que os participantes mudem de lugar, fechem os olhos novamente e pensem no complemento da frase "Eu quero..."

ATALHOS

PARA CASA:

Os alunos deverão pesquisar atalhos do teclado.

AULA 05

Rede de Computadores

(Internet/Roteadores/Servidores)

Tema:
Rede de computadores.

Materiais necessários:
Notebook, internet e projetor.

Duração:
1 hora e 30 minutos

Objetivo:
Explicar o funcionamento da internet, dos roteadores e servidores, e como esses componentes trabalham juntos.

1.1 HISTÓRIA INTERATIVA

Imagine uma grande cidade chamada Cidade Digital. Assim como na cidade onde moramos, a Cidade Digital tem casas (que representam os computadores e celulares), ruas (que são os cabos de rede), cruzamentos (os roteadores), e uma central de informação (os servidores).

Agora imagine que você quer enviar uma mensagem para um amigo que mora em outra parte da cidade. Para que essa mensagem chegue até ele, ela precisa percorrer várias ruas e cruzamentos. Quem decide o melhor caminho para essa mensagem chegar são os 'guardas de trânsito' da cidade, que são os roteadores. Esses guardas se comunicam com a Central de Informação para saber qual é o melhor caminho, e essa Central de Informação são os servidores.

Da mesma forma, quando você manda uma mensagem pelo WhatsApp ou assiste a um vídeo no YouTube, seus dados percorrem vários caminhos até chegar ao destino, passando por roteadores que direcionam as informações e servidores que armazenam esses dados.

Discussão:

Envolva os alunos perguntando: "Quem aqui já teve problemas com internet lenta? Ou já viu uma página web que não carregava?" Use essas perguntas para explicar que, às vezes, os 'guardas de trânsito' (roteadores) estão sobrecarregados, ou há um 'engarrafamento' nas ruas digitais, causando esses problemas.

Diga que durante a aula, eles vão aprender como esses 'engarrafamentos' podem ser evitados e como a Cidade Digital consegue funcionar tão bem a maior parte do tempo.

1.2. VÍDEO CURTO

Conteúdo:

Exibição de um vídeo curto, que mostre como a internet funciona de maneira visual e simples. O vídeo deve cobrir o básico de como os dados viajam pela internet, passando por roteadores e sendo armazenados em servidores.

Sugestão de vídeo: "Como funciona a Internet?" do canal GunCars.

1.3. CURIOSIDADE VISUAL

Mostre aos alunos o mapa de cabos submarinos que existem atualmente no mundo, explicando que esses cabos são grandes 'avenidas' subaquáticas que transportam dados entre países.

Sem esses cabos, a internet não funcionaria de maneira global como conhecemos. Cada vez que acessamos um site hospedado em outro país, os dados viajam por esses cabos submarinos.

Link do site: https://www.submarinecablemap.com/

INVESTIGAR

Conteúdo:

Incentivar os alunos a explorarem mais sobre o tema, utilizando ferramentas de pesquisa.

Divisão de grupos e atribuição de tópicos

Grupos:

Divida a classe em três grupos, cada um com a tarefa de pesquisar um dos seguintes tópicos ou forneça links para que os grupos encontrem informações sobre seu tópico:

Grupo 1: O que é a internet e como ela funciona?

Grupo 2: O que são roteadores e qual é a função deles nas redes?

Grupo 3: O que são servidores e como eles armazenam e compartilham dados?

AGIR

Conteúdo:

Desafiar os alunos a aplicarem o que aprenderam em um cenário prático.

Apresentação dos grupos

Cada grupo terá 5 minutos para explicar os conceitos-chave sobre os temas de forma simples e relacionar com situações do nosso dia a dia.

SAIBA MAIS

Peça para que os alunos leiam a matéria do link a seguir, reflitam e pesquisem se é possível parar a internet no Brasil. Pensem nos pontos da matéria, nas fraquezas da rede e nas formas de protegê-la. Depois, compartilhem o que pensam na aula.

https://g1.globo.com/tecnologia/noticia/2023/10/04/cabos-de-internet-no-mar-entenda-como-a-internet-chega-ate-voce-e-se-ha-risco-de-uma-acao-derrubar-a-conexao-no-brasil.ghtml

MÓDULO 2

FIGMA

Neste módulo, os alunos aprenderão conceitos básicos de design e prototipagem web, ao mesmo tempo em que desenvolvem uma visão empreendedora. Ao final, espera-se que eles dominem o uso da plataforma Figma e apresentem, durante a revisão do módulo, um protótipo de site acompanhado de um pitch, ambos desenvolvidos com base na ideia de negócio escolhida.

AULA 01

Introdução a Figma

Materiais necessários:
Notebook, internet.

Duração:
1 hora e 30 minutos

Objetivo:
Apresentação e ambientação dos alunos com a plataforma de Design Figma.

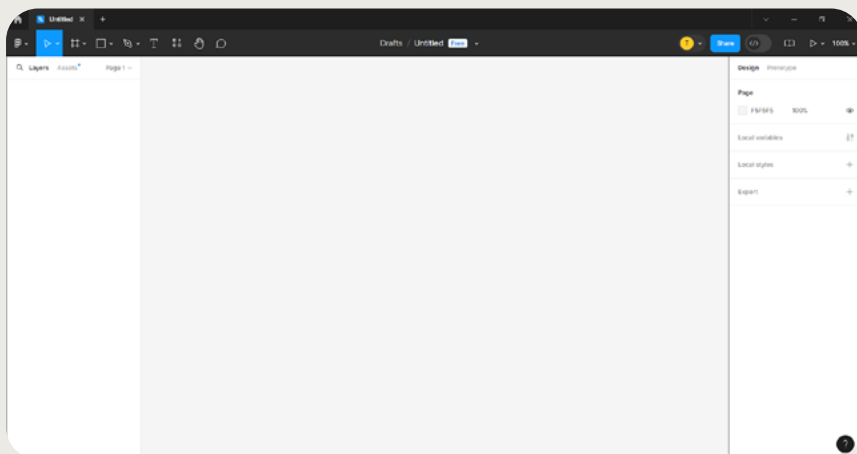
ENGAJAR

1. Apresente o que é o Figma aos participantes

Texto de referência:

O Figma é uma ferramenta de design que permite criar e testar páginas da web e aplicativos. Com o Figma, você pode trabalhar junto com outras pessoas em tempo real, o que facilita a colaboração. Ele tem recursos para fazer layouts, criar botões e simular como o site ou app vai funcionar, ajudando a visualizar tudo antes de ser feito de verdade. É uma ótima maneira de ver suas ideias ganhando vida e receber feedback de amigos ou colegas.

2. Iniciando o Figma

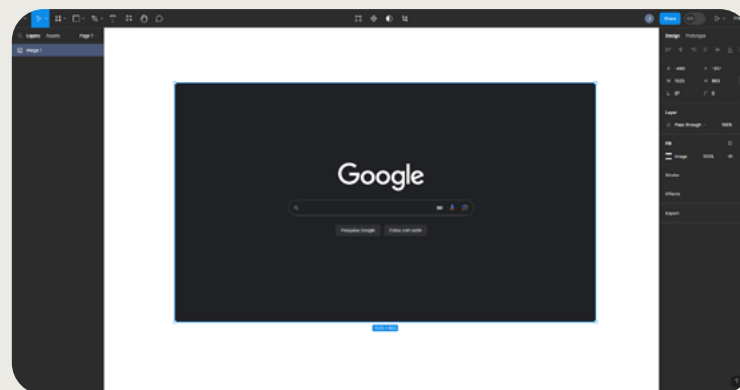
a) Conhecendo a página inicial

Na seção principal do Figma, ao estarmos conectados à plataforma, há diversas opções para explorar. Na barra de menu lateral esquerda, encontramos nossas equipes ("Your teams") e nossos arquivos de rascunho ("Draft"). É possível também marcar arquivos como favoritos e, além disso, criar equipes.

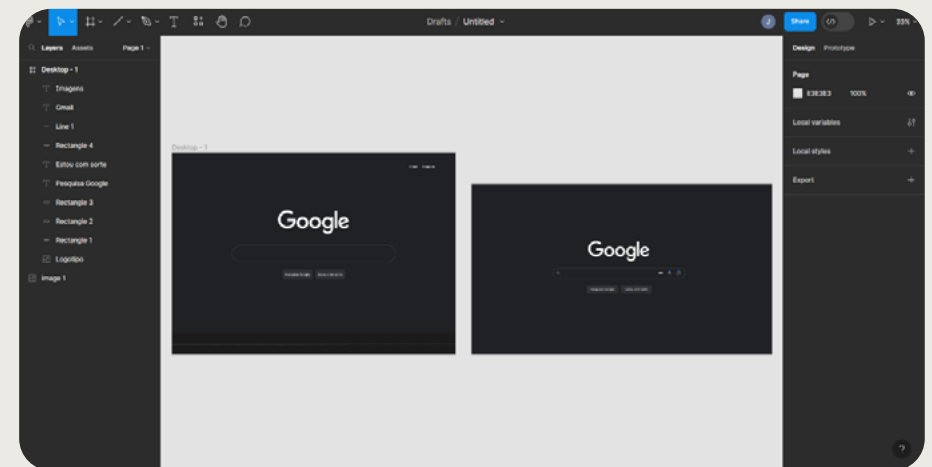
Na parte superior da interface, temos a opção "New design file" para iniciar a criação de um novo arquivo de design, seguindo a mesma abordagem utilizada anteriormente.

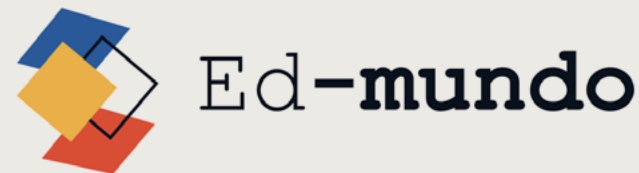
INVESTIGAR

Primeiro o(a) reaplicador(a) deve criar a conta na plataforma com o e-mail que foi criado para a turma no primeiro módulo.

Agora explore a plataforma com os alunos explicando como é possível recriar uma página como a do Google. Peça para que os alunos acompanhem enquanto você explica a atividade.

1. Para começar, cole uma captura de tela na página inicial do Figma em uma página em branco de um novo projeto. O objetivo é recriar essa página.

2. Em “Region tools”, crie um frame com as dimensões de desktop, selecionada no menu lateral direito onde vamos reproduzir a página do Google.
3. Utilize a ferramenta conta-gotas na aba de preenchimento para selecionar o cinza de fundo da página do Google e preencher o frame no Figma (localizada na aba de preenchimento (Fill) do menu lateral direito).
4. Retorne à página inicial do Google e arraste o logotipo para a área de trabalho no Figma.
5. Arraste o arquivo .png criado para o Figma e ajuste seu tamanho conforme necessário.
6. Crie o campo de pesquisa do Google com a ferramenta “Shape tools”, alinhando-o usando o recurso de alinhamento no Figma.
7. Modifique a borda do campo, ajustando a cor na seção “Stroke” no menu lateral direito.
8. Arredonde as extremidades utilizando o recurso “Corner radius” no topo do menu à direita.
9. Utilize a ferramenta de texto para inserir “Pesquisa Google” com tamanho de fonte de 16px.
10. Crie uma caixa com a ferramenta “Shape tools”, preenchendo-a com a cor correspondente ao botão na imagem de referência usando a ferramenta conta-gotas.
11. Se necessário, ajuste a ordem das camadas no menu lateral esquerdo para garantir que a camada de texto esteja à frente da camada do retângulo.
12. Ajustar o texto a 12 pixels de cada lado do botão, centrando-o. Usar a tecla “Alt” para visualizar a distância entre elementos.
13. Copiar e colar o botão à direita, alterando o texto para “Estou com sorte”. Utilize recursos de alinhamento do Figma ao movimentar elementos.
14. Adicionar uma barra retangular à base do frame, definindo sua cor com a ferramenta “Conta-Gotas”.
15. Incluir uma linha divisória usando a ferramenta “Line”, alinhando-a com o auxílio do Figma.
16. Adicionar os links “Gmail” e “Imagens” no canto superior direito para concluir a recriação da página inicial do Google.

Página feita

Peça para os alunos fazerem a logo do Edmundo. Para essa atividade, os alunos usarão as ferramentas Forma e Conta-gotas.

Ed-mundo

AGIR

AULA 02

Ideia de Negócios

Materiais necessários:
Notebook, internet, folha de resposta e lápis/caneta e imagens impressas.

Duração:
1 hora e 30 minutos

Objetivo:
Identificar a primeira proposta de negócio de cada participante durante a formação.

ENGAJAR

O(a) reaplicador(a) deve iniciar perguntando aos participantes o que eles entendem como "Valor".

Apresente um novo conceito sobre valor, diferente do valor financeiro, ressaltando que muitos produtos, serviços, empresas, marcas e instituições, presentes no dia a dia, transmitem algum tipo de valor. Isso faz com que os seus clientes/usuários se conectem com a proposta;

Depois apresente três imagens com produtos, situações ou problemas da comunidade e pergunte como os participantes se sentem ao ver as imagens (exemplo: mostrar um espaço verde com alguma comida; foto de uma pessoa utilizando uma roupa ou produto de uma marca famosa; utilizar uma imagem que apresenta um problema social); Em seguida, questionar quais são os valores pessoais que cada participante considera na hora que vai observar a imagem, se é algo que incomoda, algo que o tranquiliza, algo que o inspira, dentre outros;

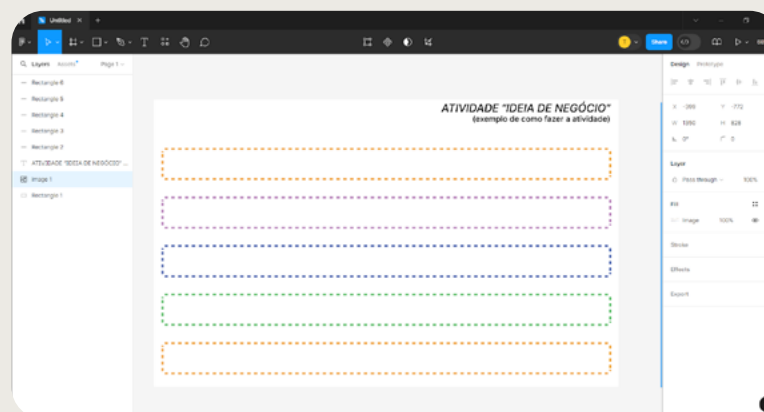
Finalizar a discussão informando que a Proposta de Valor é um elemento fundamental na hora de pensar em uma ideia de negócio, pois representa o motivo da sua existência ou aquilo que se deseja transmitir;

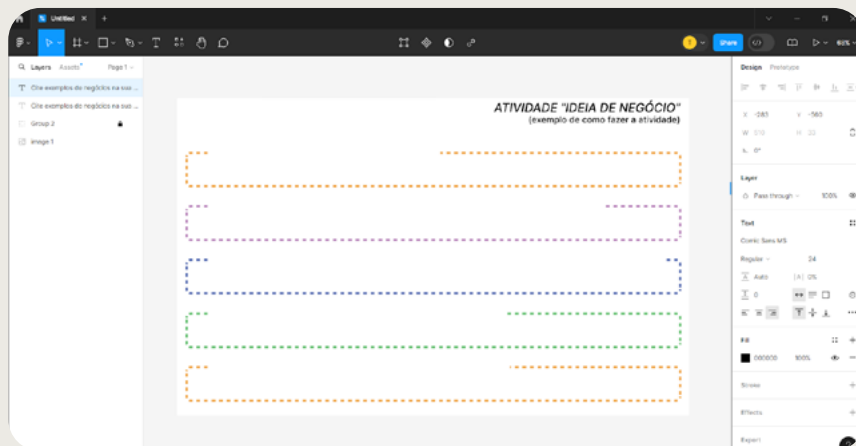
INVESTIGAR

Após a reflexão inicial, peça aos alunos para acessarem o Figma e criem um novo arquivo. Em seguida, apresente a tela da atividade "Ideia de Negócio" e pergunte como eles recriariam essa imagem no Figma. Solicite que identifiquem quais elementos aprendidos na aula anterior podem ser utilizados e quais ainda não foram abordados. Após esse momento inicial, os alunos devem recriar a tela.

Proposta de solução

Criar retângulos e aplicar as cores e as bordas tracejadas mudando o "stroke styles" para "dash" ao invés de "solid".

Adicionar o texto no canto superior direito usando uma fonte parecida. Reproduzir as caixinhas de perguntas de acordo com as cores já existentes.

Por fim, reproduzir as perguntas tentando usar uma fonte parecida.

Adicione detalhes onde ficarão as perguntas.

Depois da construção da página da atividade "Ideia de Negócio", o aluno deverá responder de acordo com o negócio escolhido.

AGIR

AULA 03

Sonhos e Problemas

Materiais necessários:
Notebook

Duração:
2 horas

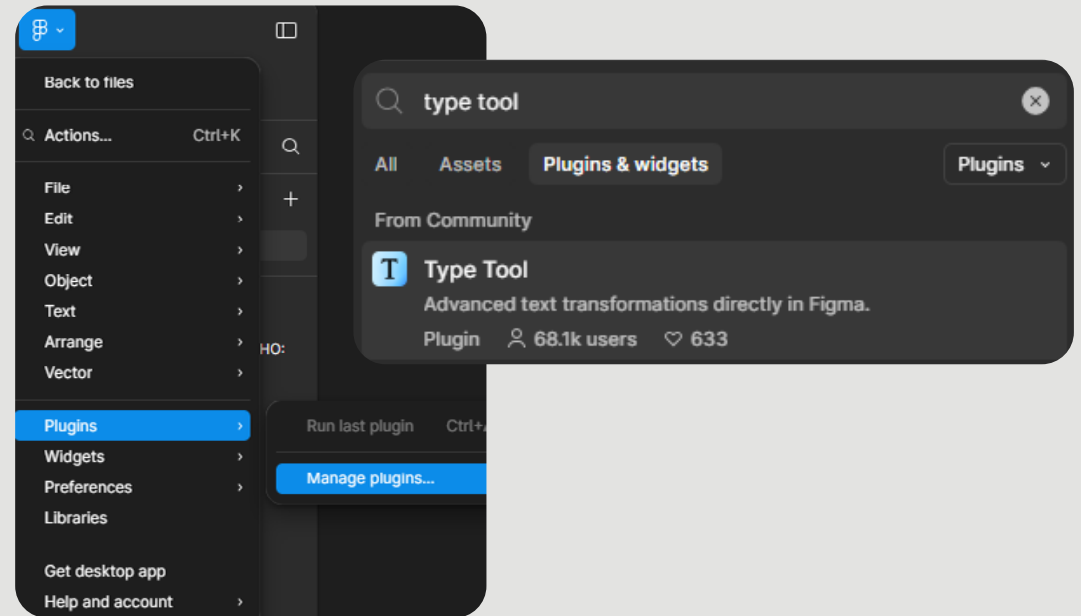
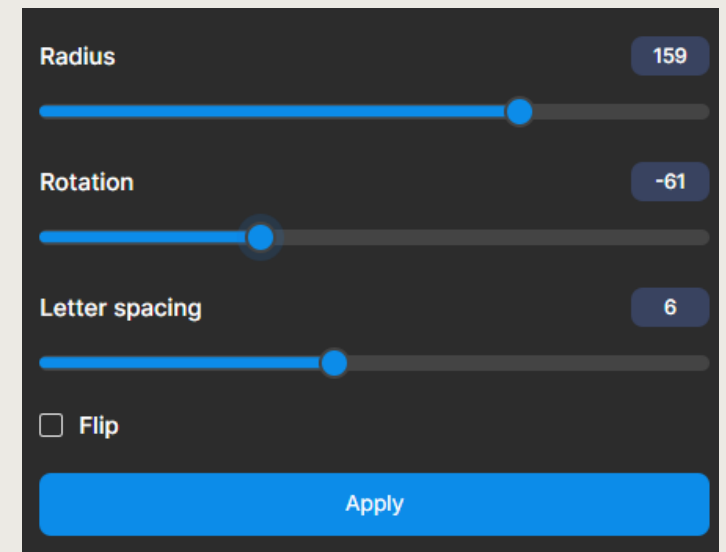
Objetivo:
Ensinar o que são e como estalar os plugins do Figma e realizar a atividade sonhos e problemas.

ENGAJAR

Primeiro apresente sobre o plugins no Figma.

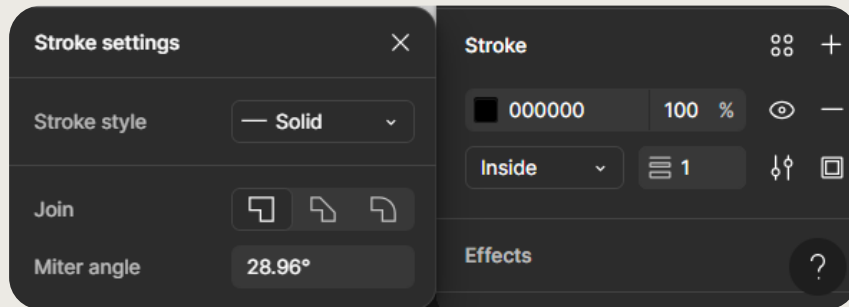
No Figma, **plugins** são extensões que aumentam a funcionalidade da plataforma e facilitam as tarefas dos usuários permitindo que melhorem seus fluxos de trabalho.

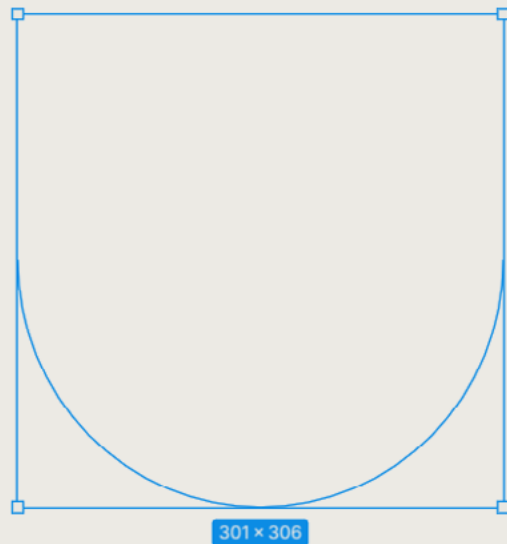
Nessa aula, iremos aprender um pouco sobre o **TypeTool**, que é usado para criar e editar textos dentro de designs, podendo aplicar rotação e angulação no texto.

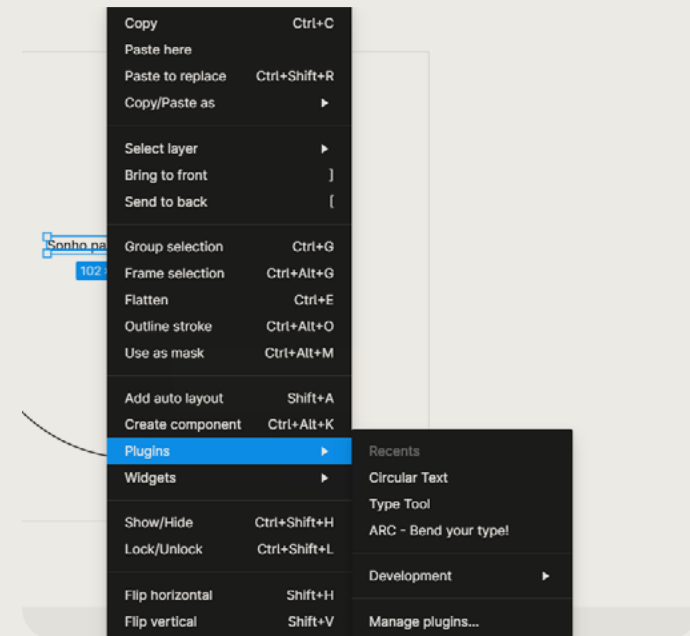
Para utilizarmos plugins é preciso acessar a barra de menu principal, ir até a seção plugins e acessar a seção "Manage Plugins". Logo após, pesquise pelo plugin "Type Tool".

Ao aplicar o plugin, temos um menu com opções para personalizar textos aplicando espaçamento e angulações nos textos.

INVESTIGAR

Peça para que os alunos explorem a ferramenta dessa aula e da aula anterior tentando recriar as seguintes imagens.

Sugestão de solução

- Personalize as cores e as bordas mudando o "stroke styles" para "dash" ao invés de "solid", como visto na aula 03.

- Para criar as circunferências você pode utilizar a forma de elipse e definir suas dimensões iguais.

- Ao clicar duas vezes é possível ajustar a forma, até que fique da forma mais próxima possível.

- Por fim, crie um texto e utilize o "type tool" clicando com o botão direito e indo na seção de plugin:

AGIR

Após a conclusão dos alunos o(a) reaplicador(a) deverá iniciar a atividade sonhos e problemas, questionando os participantes o que significa "Sonhos" e o que significa "Problemas", na concepção deles;

Em seguida, deve-se explicar o motivo de utilizar essas duas palavras em conjunto e o porquê de ambas se conectarem, uma vez que se utiliza o argumento da não realização de um determinado sonho devido a problemas identificados;

Após a reflexão, peça para que os alunos pensem em "Sonhos" por meio de três perspectivas: para a família, para a comunidade e para o país;

Após a conclusão das primeiras perguntas, indica-se que sejam respondidos nas mesmas perspectivas, os problemas que o participante identifica como sendo enfrentados;

Com a finalização da primeira etapa de perguntas, deve-se questionar "Qual é o seu sonho pessoal?", destacando que esse sonho pode ser diferente dos três tipos de sonhos já relatados;

Na sequência dos questionamentos, pergunta-se "Qual problema que você enfrenta que faz com que seu sonho pessoal não seja possível nesse momento?";

Com o intuito de conectar à atividade "Quem é você?", pede-se para responder à pergunta: Quais são as coisas que te motivam, te inspiram, ou que você gosta muito de fazer?

Ao considerar todas as respostas anteriores, deve-se orientar que os participantes pensem e respondam: Como você pode utilizar suas habilidades, coisas que te motivam e inspirações, para solucionar o problema que impede que o seu sonho pessoal se realize?

Ressalta-se que a apresentação das respostas é livre, ou seja, quem se sentir à vontade poderá apresentar, porém, destaca-se que a importância da atividade é pessoal.

AULA 04

Mapa conceitual

Materiais necessários:
Notebook, internet, folha de resposta e lápis/ caneta e imagens impressas.

Duração:
1 hora e 30 minutos

Objetivo:
Reforçar a prática com o Figma além de refletir sobre a proposta de solução /ideia de negócio de forma ampla e considerando outros aspectos

ENGAJAR

Para iniciar essa atividade, o(a) reaplicador(a) deverá pedir que os participantes retomem o problema identificado nas atividades anteriores, bem como reflitam sobre a proposta de solução que já estão desenvolvendo;

INVESTIGAR

Após a primeira reflexão, vamos pedir para que os alunos repliquem no Figma a folha da atividade “Mapa Conceitual” utilizando as ferramentas das atividades anteriores.

Como funcionará a ideia/solução?

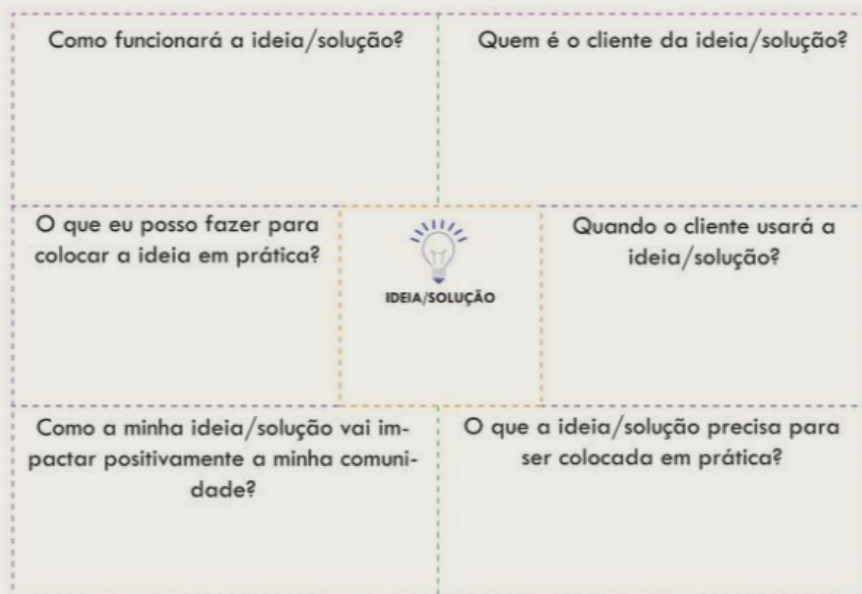
Quem é o cliente da ideia/solução?

O que eu posso fazer para colocar a ideia em prática?

IDEIA/SOLUÇÃO

Quando o cliente usará a ideia/solução?

Como a minha ideia/solução vai impactar positivamente a minha comunidade?

O que a ideia/solução precisa para ser colocada em prática?

Obs.: Existem várias formas de replicar o modelo acima. A seguir, está o passo a passo de uma das formas de como replicar a folha da atividade.

1. Inicie com uma forma retangular no centro da página e com as ferramentas de stroke estilize a borda para pontilhada e altere suas cores.

Stroke settings

Stroke style · Dash

Dash 5

Gap 5

Dash cap

Join

Miter angle 28.96°

Stroke

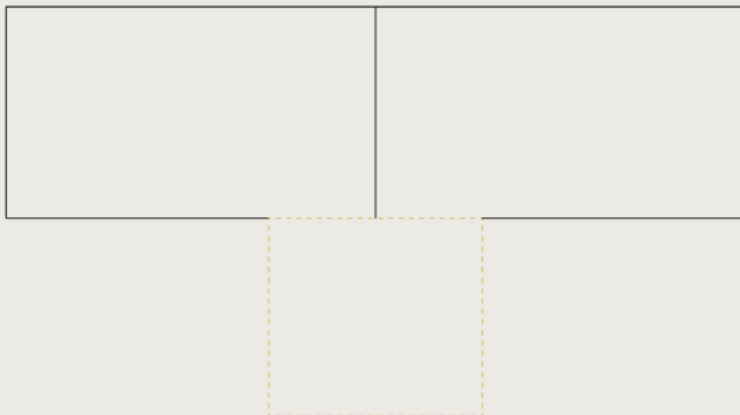
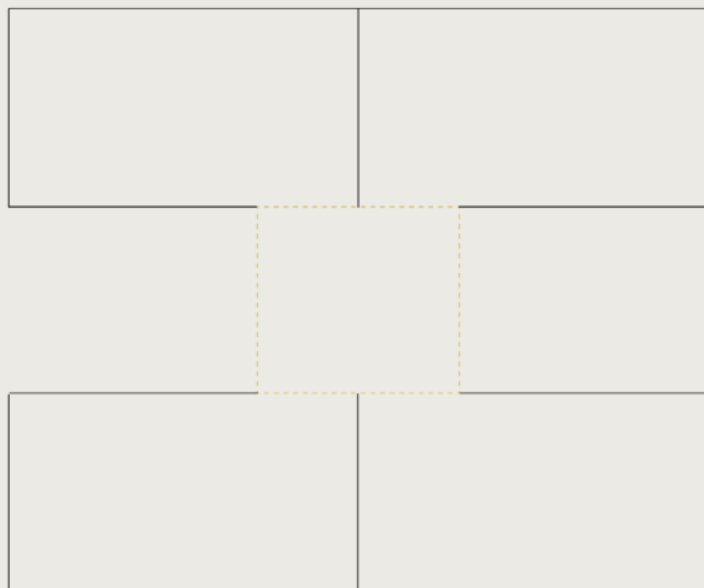
E0CE64 100 %

Inside

Effects

Export

2. Faça uma linha na face superior retângulo e nos vértices superiores.

3. Faça linhas conectando as linhas que você construiu no passo anterior.

4. Você pode selecionar as linhas que criou, copiar e girar para espelhar o desenho que já foi feito.

3. Por fim, conecte as linhas do desenho e aplique as formatações de borda utilizando as cores de preferência (assim como feito no passo 1). Além disso, preencha cada retângulo com as perguntas que serão respondidas no AGIR.

Como funcionará a ideia/solução?

Quem é o cliente da ideia/solução?

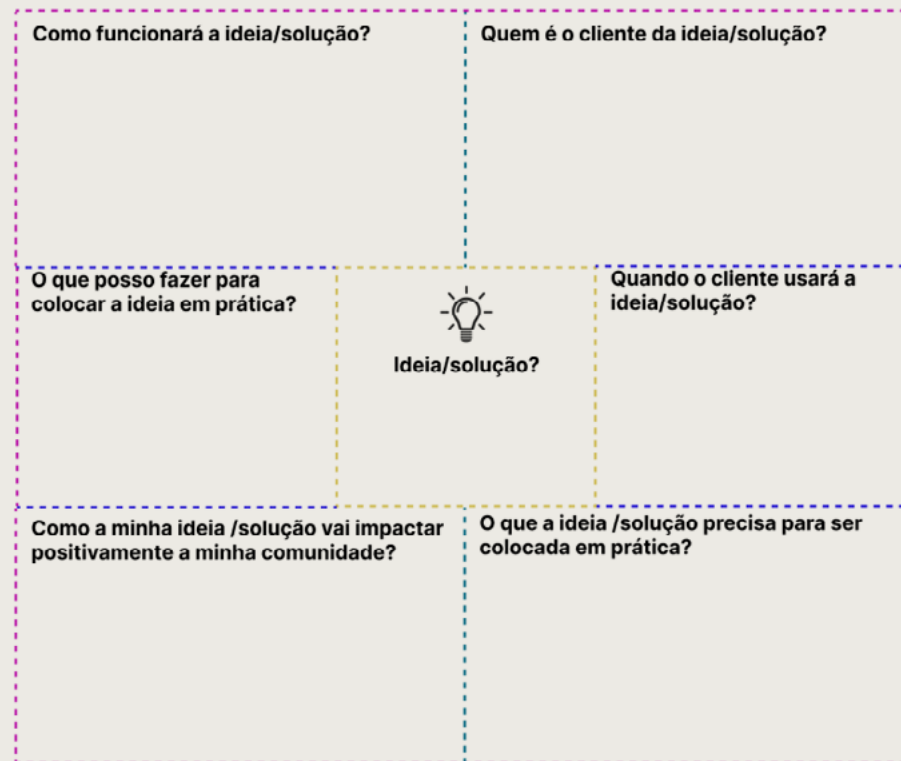
O que posso fazer para colocar a ideia em prática?

Ideia/solução?

Quando o cliente usará a ideia/solução?

Como a minha ideia /solução vai impactar positivamente a minha comunidade?

O que a ideia /solução precisa para ser colocada em prática?

AGIR

1. Recomendar que os participantes comecem pelo centro da folha, descrevendo a “ideia de negócio/proposta de solução”;
2. Em seguida, o participante deverá responder: “Como funcionará a ideia/solução?”, em que se descreverá, de que forma a ideia acontecerá na prática;
3. A etapa posterior é responder: “Quem é o cliente da ideia/solução?”, recuperando o que foi utilizado na atividade Mapa de Empatia;
4. A quarta pergunta a ser respondida é: “Quando o cliente utilizará a ideia/solução?”, especificando a frequência ou período de utilização;
5. Logo após, o(a) reaplicador(a) deverá solicitar que os participantes respondam: “O que a ideia /solução precisa para ser colocada em prática?”, sendo importante listar as etapas necessárias para que a proposta aconteça;
6. A penúltima pergunta é “Como a minha ideia/solução vai impactar positivamente a minha comunidade?”, destacando os benefícios da realização (implementação da ideia;
7. Por fim, a pergunta a ser respondida é: “O que eu posso fazer para colocar a ideia em prática?”, destacando o que cada pessoa vinculada à ideia pode fazer para que ela aconteça;
8. Para finalizar, o(a) reaplicador(a) deverá perguntar o que os participantes acharam da experiência de refletir sobre os pontos apresentados na atividade e abrir o espaço para quem quiser compartilhar, ressaltando que a atividade não precisa ser entregue, pois é destinada à cada participante, auxiliando no desenvolvimento na sua ideia.

AULA 05

Mapa da Empatia

Materiais necessários:
Computador, internet e projetor.

Duração:
2 horas

Objetivo:
Apresentar a ferramenta “pen” no Figma. Definir e compreender o perfil dos clientes para os quais se deseja projetar uma solução.

ENGAJAR

O(a) reaplicador(a) deve começar a aula explicando um pouco sobre a ferramenta Pen no Figma.

No Figma, a ferramenta **Pen** é uma ferramenta de desenho usada para criar formas, curvas, linhas e caminhos personalizados com precisão. Ela permite desenhar objetos geométricos mais complexos, definindo pontos e conectando-os com linhas retas ou curvas.

O(a) reaplicador(a) pode utilizar o exemplo da primeira aula para mostrar como a "Pen" pode ser utilizada em conjunto com as ferramentas de preenchimento e conta-gotas para replicar a logo do Ed-mundo através de um print.

INVESTIGAR

Deve-se apresentar o Mapa de Empatia, destacando brevemente os seus principais elementos como a caracterização da pessoa que está no meio e o que deve ser inserido em cada espaço.

Pedir aos alunos que abram o Figma e criem um arquivo em 'New Design File'. Eles terão que reproduzir a imagem abaixo.

O que a pessoa sente?

O que a pessoa pensa ou escuta?

O que ela gosta de fazer?

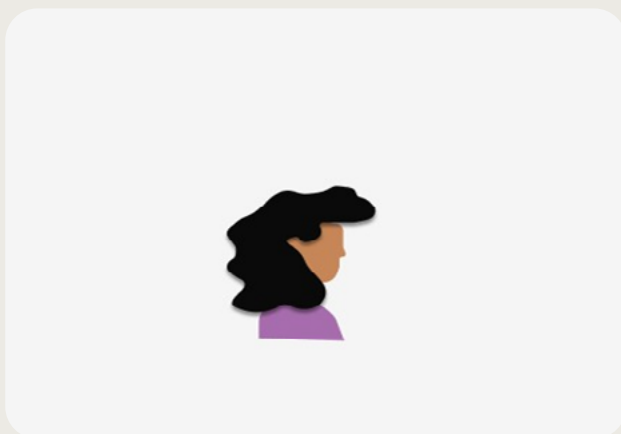
O que ela vê?

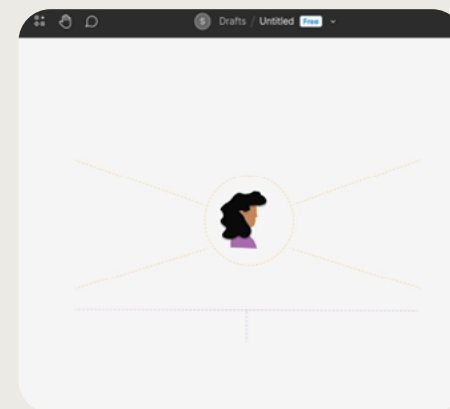
Dores/Obstáculos

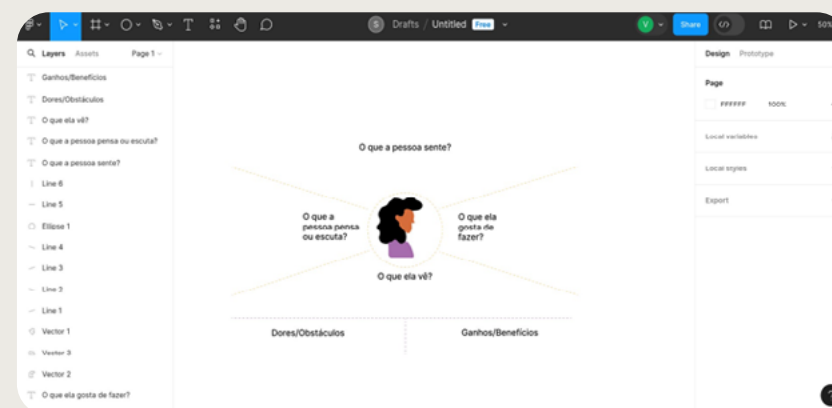
Ganhos/Benefícios

Sugestão de solução:

1. Após criar um novo arquivo, adicionaremos a imagem, selecionaremos a função “Pen” e contornaremos a primeira parte do desenho. Para visualizar melhor, use o zoom.

Após contornar todas as partes do desenho, tire a borda e adicione a cor. Ficará assim... Resultado:

3. Adicionar “Ellipse” ajustar o tamanho, retirar fundo/cor, mudar a borda para “Dash” no tamanho “6” e adicionar cor;

4. Adicionar o “Line” para fazer as linhas, mudar pra “Dash” no tamanho “6”, adicionar cor;

7. Adicionar “Text” para fazer as perguntas;

Resultado final.

OBS.: O professor/reaplicador deverá dar dicas ou mostrar aos alunos onde ficam todas as funções.

Após a conclusão dos alunos, o(a) reaplicador(a) deverá iniciar a atividade Mapa de Empatia.

Perguntar se os participantes conhecem o significado de empatia e se podem dar um exemplo prático do que seria. Permitir que cada um possa expressar sua opinião.

Empatia é a capacidade de perceber e compreender os sentimentos e emoções de outras pessoas. Envolve a habilidade de se colocar no lugar do outro, percebendo suas experiências e perspectivas, o que permite uma conexão emocional mais profunda. A empatia pode levar a comportamentos altruístas e ao fortalecimento de relacionamentos, pois promove compreensão e apoio mútuos.

Deve-se associar o conceito de empatia com a proposta dos projetos, se possível citando exemplos, e explicar que um dos principais determinantes de uma solução é a pessoa que irá utilizá-la, ou seja, o usuário/a cliente.

Conclua com uma reflexão sobre a importância de integrar a empatia na criação de negócios, destacando como um entendimento profundo das necessidades e sentimentos do cliente pode resultar em soluções mais eficazes e inovadoras.

Após concluído a montagem da atividade, recomenda-se que os alunos comecem respondendo: “o que a pessoa sente?”, considerando suas características e buscando se colocar em situações que a própria pessoa possa ter vivenciado ou está vivenciando.

Logo após, indique que a próxima pergunta norteadora deve ser: “o que a pessoa pensa ou escuta?”, refletindo com base nas pessoas que estão em volta ou que convivem com a sua personagem.

A terceira pergunta é: “o que ela gosta de fazer?”, e deve ser respondida considerando os prováveis interesses da pessoa descrita.

Em seguida, a pergunta deve ser: “O que ela vê?”, com base no ambiente em que a pessoa mora ou frequenta ou até mesmo nos elementos que a mesma mais direciona seu olhar, pode ser, inclusive, seus incômodos.

Por fim, peça que os participantes observem os dois quadrantes finais das ferramentas que destacam dores e ganhos. Com isso, deve-se explicar que as respostas desses dois itens devem ser pautadas, associando-as à proposta de solução visualizada pela equipe. Ou seja, em “Dores/Obstáculos”, deve-se refletir sobre: “Quais as dores essa pessoa tem para consumir o produto/serviço proposto?”, considerando as barreiras, dificuldades e obstáculos necessários para que isso ocorra. Em seguida, no tópico “Ganhos/Benefícios”, serão pensados quais ganhos a pessoa tem ao consumir/acessar/utilizar esse produto/serviço/solução, ou seja, o que a pessoa ganharia com isso?

Ação do(a) Reaplicador(a)

Reforçar que o direcionamento para o preenchimento das informações deve ser sempre em relação à pessoa que vivencia o problema que a proposta da equipe busca solucionar. Responder dúvidas ao longo da aplicação da ferramenta.

AULA 06

Jornada do usuário

Materiais necessários:
Computador, internet e projetor.

Duração:
1 hora e 30 minutos

Objetivo:
Entender o que é a jornada do usuário e criar a jornada dos possíveis usuários da sua ideia de negócio.

ENGAJAR

Apresentar um exemplo prático do que seria o caminho percorrido pelo usuário/cliente desde o problema até chegar a utilizar um produto/serviço.

Para fortalecer o debate, solicita-se que os participantes comentem o que entendem por tecnologia e de que forma se relacionam com a mesma.

Após ouvir a resposta de todos os participantes, o(a) reaplicador(a) deverá comentar que muitos produtos e serviços são encontrados por meio da internet. Assim como questões como logística, desenvolvimento de novos produtos e até as vendas são possibilitadas pelo uso da tecnologia, é preciso considerar esses elementos nas ideias que estão surgindo agora.

INVESTIGAR

Peça para os alunos abrirem o Figma, e criarem um arquivo de Design. Eles deverão reproduzir a imagem abaixo para realização da atividade.

Antes

Durante

Depois

Como a tecnologia pode ser utilizada na minha ideia de negócio?

Proposta de solução:

1. Criar um arquivo na opção "New design file";
2. Selecionar a "Pen", e iniciar os desenhos;
3. Selecionar o primeiro vetor e alterar sua cor para laranja;
4. Selecionar o segundo vetor e alterar sua cor para azul;
5. Selecionar o terceiro vetor e alterar sua cor para roxo;
6. Selecionar o quarto vetor e alterar sua cor para azul;
7. Selecionar a ferramenta "Text" para adicionar os textos conforme a imagem.

AGIR

A atividade será respondida em duas etapas conforme a divisão apresentada no modelo elaborado no Figma.

Para começar, explica-se a importância de reconhecer o trajeto vivenciado pelo usuário/cliente e a forma que o mesmo se relaciona com a proposta de solução, pedindo para que os participantes observem que na folha há três espaços: antes, durante e depois.

Esses tópicos correspondem aos períodos vivenciados pelo cliente da sua proposta:

- O primeiro indica o que as pessoas faziam antes de existir o seu serviço/produto/ideia;
- O segundo representa o que as pessoas fazem durante o uso do serviço;
- O terceiro representa o que elas fazem depois de utilizar.

Ao concluir essa etapa, abre-se o espaço para que os participantes apresentem o que já definiram;

Para finalizar, cada participante deverá responder abaixo “Como a tecnologia pode ser utilizada na minha ideia de negócio?”.

AULA 07

Mapa da empatia + Plugin icons8

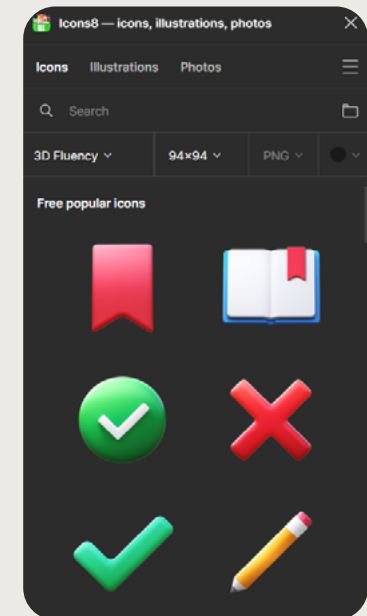
Materiais necessários:
Computador, internet e projetor.

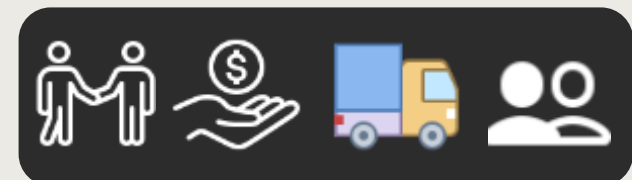
Duração:
2 horas

Objetivo:
Introduzir o Modelo Canvas para estruturar ideias de negócio de forma prática.

ENGAJAR

O(a) reaplicador(a) deve iniciar a aula explicando aos participantes sobre o plugin icons8 e a importância de ícones como ferramentas visuais.

O plugin Icons8 é uma ferramenta que integra uma biblioteca de ícones no Figma. Ícones são ferramentas visuais essenciais em design, melhoram a comunicação visual, facilitam a navegação e criam interfaces de usuário mais intuitivas e criativas. Como por exemplo, os botões de like em redes sociais como instagram e youtube são representados por ícones de coração ou por um "joinha".

INVESTIGAR

Mostre para os alunos esse modelo canvas e peça para que eles reflitam sobre os ícones escolhidos para cada campo e o que eles querem indicar.

Peça para os alunos recriarem o modelo canvas respeitando as ordens dos campos, mas personalizando os ícones com os recursos de ícones do plugin icons8.

Exemplo de plugins que podem ser usados.

A G I R

Explique brevemente o que é o Modelo de Negócio Canvas e sua função de ajudar a organizar visualmente os principais elementos de um negócio em nove blocos: Proposta de Valor, Clientes, Canais, Relacionamento com Clientes, Fontes de Receita, Atividades Principais, Parcerias Principais, Recursos Principais e Estrutura de Custos.

Mostre para os alunos esse modelo canvas e explique cada um dos campos:

PROPOSTA DE VALOR

É uma frase que representa a sua ideia de negócio, algo que você gostaria de levar ao seu cliente, mas que não é a sua solução, seria algo à mais, reforçando a ideia da proposta de solução criada ao longo do projeto. Por exemplo: deixar o cliente mais feliz, proporcionar a melhor experiência de utilização, frases desse tipo.

CLIENTES

O perfil de clientes que o participante criou no Mapa de Empatia. Canais: Formas de comunicação/interação com seu cliente, como redes sociais, delivery, dentre outras.

RELACIONAMENTO COM OS CLIENTES

Inserir informações sobre a sua forma de manter o contato com o cliente, por exemplo, se a proposta for um site, terá um espaço para tirar dúvidas? Ocorrerão visitas presenciais? O contato seria por meio de reuniões virtuais? Esses e outros pontos devem ser descritos nessa parte.

FONTES DE RECEITA

Relatar de que forma entrará dinheiro no seu negócio, ou seja, o que será comercializado, se são produtos, serviços, publicidade, dentre outros.

ATIVIDADES PRINCIPAIS

Inserir todas as atividades que o negócio vai desenvolver, conectando com a atividade "Mapa Conceitual".

PARCERIAS PRINCIPAIS

Parceiros do seu negócio, lembrando do que já foi trabalhado na atividade "Quem me apoiará?".

RECURSOS PRINCIPAIS

As informações sobre tudo o que você precisa para realizar suas atividades, tanto recursos financeiros, como materiais ou pessoas para colaborar.

ESTRUTURA DE CUSTO

Tudo precisará comprar ou contratar para que seu negócio aconteça, ressaltando que esses itens foram observados nas atividades "Mapa Conceitual" e "Custos e Receitas".

Após a explicação dos tópicos, peça para que os alunos preencham os campos do canva.

AULA 08

Elementos de um site

Materiais necessários:

Computador, internet e projetor.

Duração:

1 hora e 30 minutos

Objetivo:

Ensinar os conceitos básicos de elementos de um site.

ENGAJAR

O(a) reaplicador(a) deverá apresentar alguns dos elementos mais básicos que geralmente um site tem.

Exemplo:

Cabeçalho:
Parte superior do site e normalmente inclui logotipo, menu de navegação e barra de busca: Para facilitar a localização de conteúdo dentro do site.

Corpo de conteúdo:
O corpo de conteúdo é onde está a maior parte das informações. Ele pode incluir título e subtítulo, parágrafo, imagens, vídeos ou ilustrações.

Rodapé:
Parte inferior da página e contém informações complementares, como informações de contato e redes sociais.

INVESTIGAR

Após a apresentação dos elementos básicos de um site, apresente a eles o seguinte protótipo feito no Figma.

Faça os seguintes questionamentos aos alunos:

- É possível identificar todos os elementos básicos apresentados anteriormente?
- O protótipo da página mostra bem o propósito do Ed-mundo?
- As cores escolhidas para esse protótipo fazem sentido e ficam visualmente agradável?
- Quais elementos poderiam ser inseridos nessa página?

AGIR

Peça para os alunos recriarem a página acima, adicionando o footer e elementos que os alunos sugeriram durante a sessão de investigação.

AULA 09

Elementos de um site (formulários)

Materiais necessários:
Computador, internet e projetor.

Duração:
1 hora e 30 minutos

Objetivo:
Criar um login de acesso.

ENGAJAR

O(a) reaplicador(a) deverá iniciar a aula falando sobre a importância de formulário em sites.

Formulários são componentes essenciais que realizam uma interação entre o usuário e o site como no seguintes exemplos:

Dados de contato:	Nome, e-mail, telefone etc., para comunicação ou envio de newsletters.
Informações para cadastro:	Formulários de registro permitem a criação de contas, oferecendo acesso a recursos restritos.
Feedback:	Comentários, sugestões ou críticas que ajudam a melhorar produtos, serviços ou a própria experiência no site.

INVESTIGAR

De acordo com a parte inicial explicada, peça para que os alunos deem exemplos de formulários que eles preenchem no dia a dia, seja em sites ou aplicativos.

Alguns exemplos de formulários:

- Login em redes sociais;
- Formulários de inscrição;
- Formulários de feedback;
- Formulários de recuperação de senha.

A seguir, apresente o seguinte formulário aos alunos:

Olá, ED-Mundo!

Entrar

Recupere sua senha clicando aqui

Ed-mundo

Crie sua conta agora mesmo!

Usuário

Email

Senha

Criar

Mostre a eles que alguns efeitos de sombra foram adicionados aos botões. Peça para que eles pesquisem como aplicar esses efeitos no Figma.

AGIR

Peça para que os alunos recriem o formulário que foi apresentado na sessão anterior, peça para que também insiram ícones nos campos de usuário, e-mail e senha.

Obs.: Caso os alunos terminem antes do final da aula, peça para que eles tentem recriar a tela de login, mas dessa vez para uma aplicação mobile.

AULA 10

Pitch

Materiais necessários:
Computador, internet, projetor, lápis/caneta, folhas A4.

Duração:
1 hora e 30 minutos

Objetivo:
Construir uma apresentação da ideia de negócio/ proposta de solução.

O(a) reaplicador(a) deverá explicar que o pitch é a apresentação que representa a finalização do ciclo do projeto, com o intuito de receber feedbacks e implementar melhorias nas ideias.

Explique como será a dinâmica de apresentação e avaliação do Pitch para os alunos.

Para apresentar o pitch, o participante terá de 3 a 5 minutos, conforme definição do(a) reaplicador(a) ao analisar a situação de cada turma.

As apresentações serão assistidas por 3 avaliadores, que irão considerar aspectos como inovação, potencial de escala, equipe, próximos passos da ideia, impacto na comunidade, sustentabilidade financeira e uso da tecnologia digital.

Após cada apresentação, será aberto o espaço para que cada avaliador possa dar feedbacks, dar sugestões ou tirar dúvidas. O objetivo desses comentários não é o de julgar se a ideia é boa ou ruim, mas sim, de destacar quais pontos no projeto de cada participante podem ser melhorados.

Obs.: Se a apresentação do pitch for gravada, escolha uma parede de fundo liso, para não causar distrações, e grave um vídeo de até 3 minutos com a sua apresentação construída na atividade anterior. Lembre-se de considerar o máximo de pontos que você relatou no texto do seu roteiro.

ENGAJAR

Solicitar que os participantes leiam atentamente as perguntas e as "dicas" abaixo de cada enunciado:

01 | Qual seria a frase/palavra impactante que mais representaria sua ideia de negócio?

Dica: Utilize até 7 palavras. Deve ser objetiva e chamar atenção para a essência do problema que você busca resolver com a plataforma.

02 | Se você pudesse contar em formato de história o problema que você busca solucionar com a sua ideia, como seria?

Dica: Seja sucinto e busque criar empatia com o público. Exemplos e histórias inspiradoras podem ajudar você.

03 | Qual a solução que você traz no seu negócio?

Dica: Explique o funcionamento do seu produto ou serviço.

04 | De que forma seu negócio impacta a sociedade?

Dica: Você pode comentar sobre os atores envolvidos, o local, fatores sociais e culturais, dentre outros. Cite os mais importantes.

05 Qual é o seu tamanho de mercado?

Dica: Comente seu público-alvo com clareza. Vale a pena retomar as personas que você criou no mapa de empatia e observar o seu canvas.

06 Qual o valor que sua ideia de negócio precisa para iniciar? Como ela vai se sustentar financeiramente?

Dica: Utilizar informações da planilha financeira e dos produtos que serão oferecidos pelo negócio.

07 Quais são os diferenciais do seu negócio em relação as outras iniciativas do segmento?

Dica: Comentar o que diferencia o seu negócio dos potenciais concorrentes e serviços semelhantes.

08 Como os seus produtos e serviços chegarão aos clientes? Como a tecnologia estará presente na sua ideia de negócio?

Dica: Comentar sobre os canais de comunicação, venda e divulgação.

09 Quem seria a equipe do seu negócio?

Dica: Falar o nome das pessoas e as funções.

10 Qual seria a informação final sobre o seu negócio e quais os próximos passos você acredita que seriam o caminho para execução da proposta de negócio?

Dica: Escolha uma frase curta e que seja convidativa, apresente de forma confiante os próximos passos da ideia.

Após preencher todas as questões, o(a) reaplicador(a) deverá unir todas as respostas que anotou no papel, e em uma folha em branco, transforme-as em um texto único, que servirá como roteiro para sua apresentação; 4. Deve-se destacar que os participantes conectem as respostas, sem citar o enunciado das questões, mas criando uma coerência no texto; 5. Para finalização, abre-se o espaço para que os participantes ensaiem a apresentação do pitch.

AGIR

Após preencher todas as questões, o participante deverá unir todas as respostas que anotou no papel, e em uma folha em branco, transformando-as em um texto único, que servirá como roteiro para sua apresentação.

Deve-se destacar que os participantes devem conectar as respostas, sem citar o enunciado das questões, criando uma coerência no texto.

Por fim, abre-se o espaço para que os participantes ensaiem a apresentação do pitch.

AULA 11, 12 e 13

Protótipo de site com ideia de negócio e preparação do Pitch

Materiais necessários:
Computador, internet, projetor, lápis/caneta, folhas A4.

Duração:
1 hora e 30 minutos

Objetivo:
Construir um site com os conhecimentos obtidos durante as aulas e preparar a apresentação da ideia de negócio/proposta de solução.

AGIR

Nas próximas aulas os alunos deverão realizar a construção do protótipo de um site com base na ideia de negócio que foi trabalhada durante o módulo. O site deverá conter no mínimo 4 telas sendo elas tela principal, tela de login, tela de cadastro, tela de livre escolha.

O aluno que quiser ir além poderá criar sua logo, estudar sobre sua identidade visual e fazer mais telas para seu protótipo.

O papel do reaplicador durante essas aulas é de acompanhar os alunos fazendo sugestões de melhorias e direcionando suas dúvidas.

Além do site, os alunos devem aproveitar as aulas para complementar e/ou revisar o pitch.

AULA 14

Review 02

Materiais necessários:
Projetor, lápis/caneta, computador e fichas de avaliação.

Duração:
1 hora e 30 minutos

Objetivo:
Apresentação da ideia de negócio (Pitch e Protótipo)

Nessa aula os alunos devem apresentar suas ideias de negócio e Pitch para 3 avaliadores.

A banca de avaliar aspectos como inovação, potencial de escala, equipe, próximos passos da ideia, impacto na comunidade, sustentabilidade financeira e uso da tecnologia digital.

Cada critério deve ser avaliado de 0 a 10, onde 0 significa que o aspecto está menos presente, e 10 significa que está excelente ou totalmente atendido.

Após cada apresentação, será aberto o espaço para que cada avaliador possa dar feedbacks, dar sugestões ou tirar dúvidas. **O objetivo desses comentários não é o de julgar se a ideia é boa ou ruim, mas sim, de destacar quais pontos no projeto de cada participante podem ser melhorados.**

Critério	Descrição	Pontuação (0-10)									
		Pitch 1	Pitch 2	Pitch 3	Pitch 4	Pitch 5	Pitch 6	Pitch 7	Pitch 8	Pitch 9	Pitch 10
Inovação	Quão inovadora é a ideia proposta? Traz algo novo ao mercado ou à solução de problemas existentes?										
Potencial de escala	A ideia tem potencial para crescer rapidamente? Pode ser replicada em diferentes mercados ou regiões?										
Equipe	A equipe possui as habilidades, experiência e competência necessárias para executar o projeto?										
Próximos passos da ideia	Quão claros e viáveis são os próximos passos para o desenvolvimento e execução da ideia?										
Impacto na comunidade	A solução tem um impacto positivo significativo na comunidade ou na sociedade?										

Sustentabilidade financeira	O modelo de negócios é sustentável a longo prazo? A empresa poderá gerar receita suficiente para crescer?										
Uso da tecnologia digital	A tecnologia digital é bem utilizada na solução? Proporciona eficiência, inovação ou escalabilidade?										
Design visual	A apresentação do protótipo tem um design atraente e consistente? Há equilíbrio de cores, tipografia e layout?										
Usabilidade (UX)	O protótipo do site oferece uma navegação simples e intuitiva? É fácil para os usuários interagirem com ele?										
Clareza do propósito	O protótipo reflete claramente a ideia de negócio apresentada?										
Coerência com a identidade da marca	O design do protótipo está alinhado com a identidade da marca (cores, tipografia, estilo visual)?										

Ação do(a) Reaplicador(a):

- Organizar as apresentações e suas ordens.
- Poderá utilizar um pendrive como recurso para guardar as apresentações em um único só dispositivo.

MÓDULO 3

FLUXOGRAMA E LÓGICA DE PROGRAMAÇÃO

Neste módulo, os participantes serão desafiados a resolver problemas introdutórios de lógica, utilizando o esboço de algoritmos e a implementação de soluções por meio da programação no Scratch.

Ao final do módulo, espera-se que os alunos compreendam os símbolos que compõem um fluxograma, sejam capazes de esboçar algoritmos para resolver problemas e tenham uma introdução aos princípios básicos da programação, incluindo estruturas de repetição, condicionais, variáveis e operadores.

Recomendação:

Caso seja possível, instale o aplicativo Scratch nos computadores/notebooks que serão utilizados para as aulas. Isso evitará atrasos no conteúdo por conta da conexão com a internet. O aplicativo pode ser baixado diretamente no site da plataforma scratch.

AULA 01

Resolução de problema utilizando algoritmos

Materiais necessários:
Computador, projetor e internet.

Duração:
1 hora e 30 minutos

Objetivo:
Mostrar a importância de seguir o passo a passo na ordem adequada para solucionar problemas/desafios.

ENGAJAR

O(a) reaplicador(a) deve iniciar a aula explicando o que são fluxogramas e contextualizando como eles podem ser utilizados para propor soluções para problemas do cotidiano.

Nós resolvemos diversos problemas do nosso cotidiano sem perceber e durante a resolução, seguimos um conjunto de passos para que o problema possa ser superado. Com o tempo, descobrimos as respostas e passamos a seguir os mesmos procedimentos, evitando que os problemas continuem ocorrendo. Por exemplo, todos vão à escola durante a semana. Para ir à escola, você sempre faz um conjunto de atividades até que o objetivo, que é chegar à escola, seja atingido. Esse objetivo pode ser considerado um "problema".

As notações são regras que devem ser seguidas para a criação dos fluxogramas. Essas notações são bem simples e os seus significados sempre serão os mesmos. As notações são figuras geométricas que possuem um determinado significado. As figuras abaixo demonstram o significado das principais notações que são utilizadas na criação de um fluxograma.

Terminação

Símbolo que é utilizado para indicar o início e o fim do fluxograma.

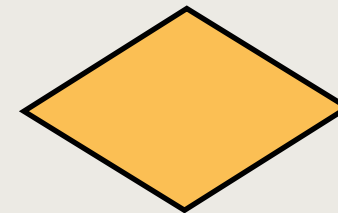
Processo

Símbolo utilizado para indicar uma atividade que deverá ser executada.

Dados

Símbolo utilizado para indicar que é necessária alguma informação para que o algoritmo continue sendo processado.

Decisão

Símbolo utilizado para indicar quando alguma decisão durante o fluxo precisa ser tomada.

Fluxo

Permite indicar qual é o fluxo (caminho) que um fluxograma adotará.

INVESTIGAR

Vamos fazer um exemplo para que os alunos compreendam melhor o assunto. O problema é: **ir à escola**. Faça anotações de etapas sugeridas pelos alunos para realizar essa atividade. Observe se a sequência apresentada corresponde ao passo a passo necessário para a resolução do problema. De forma coletiva, peça para que os alunos indiquem por meio de números, qual a ordem correta para organizar essas informações, seguindo a lógica da primeira atividade até a última. Lembre-se, as lógicas de programação precisam estar em uma ordem correta!

Exemplo de possível solução:

1. Tomar banho;
2. Almoçar;
3. Escovar os dentes;
4. Arrumar a mochila;
5. Sair de casa;
6. Caminhar até a escola.

Agora peça para que os alunos respondam no bloco de notas do computador, o passo a passo para realizar as atividades:

- Ir à mercearia comprar um refrigerante;
- Fazer um suco de laranja;
- Fazer pipoca na panela.

Exemplo de possível solução:

Ir à mercearia comprar um refrigerante:

1. Tomar banho
2. Almoçar
3. Escovar os dentes
4. Arrumar a mochila
5. Sair de casa
6. Caminhar até a escola

Fazer um suco de laranja:

1. Pegar as laranjas
2. Cortar as laranjas
3. Espremer as laranjas
4. Coar o suco
5. Colocar o suco na jarra

Fazer pipoca na panela:

1. Pegar a panela;

2. Colocar no fogão;

3. Colocar o óleo na panela;

4. Colocar o milho na panela;

5. Acender o fogo;

6. Fazer a pipoca.

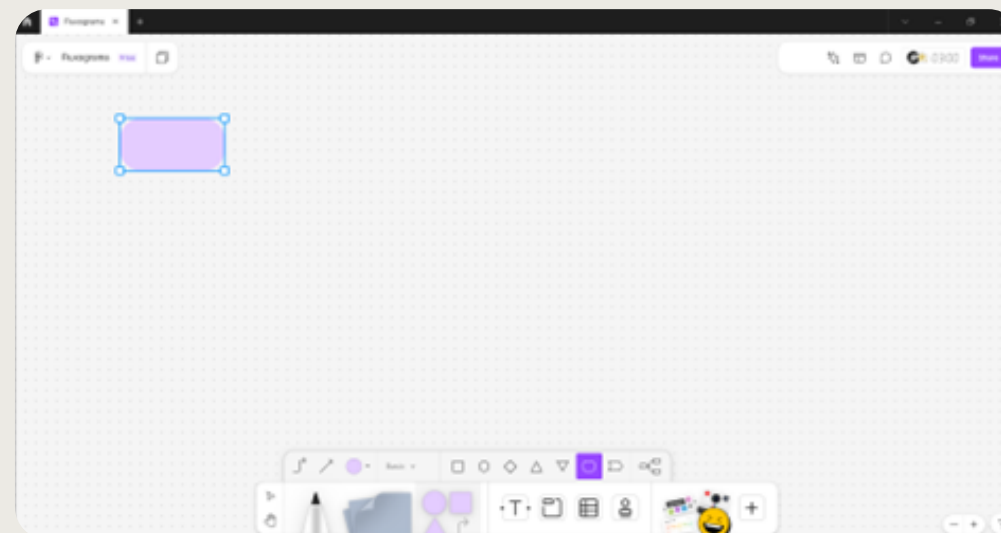
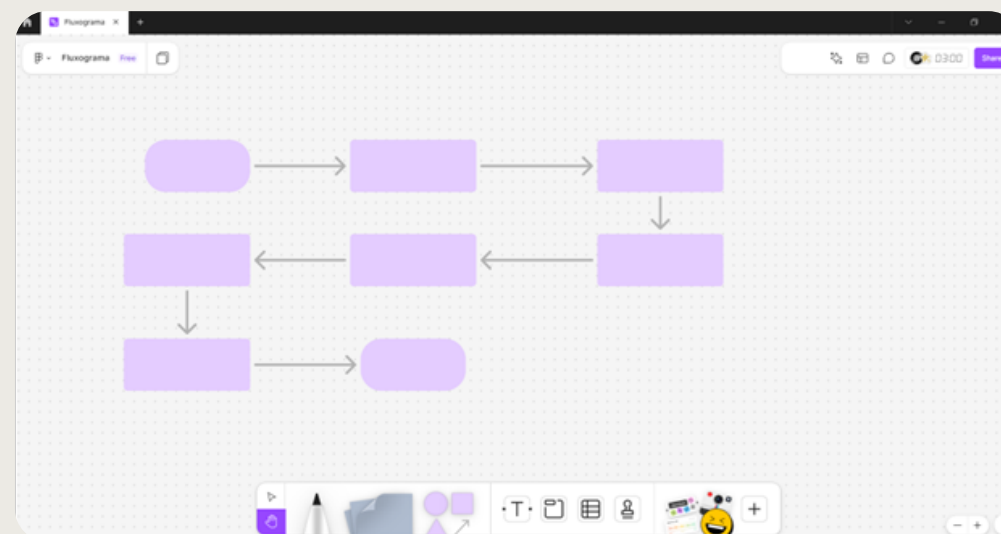
AGIR

Para que os alunos fixem e explorem o conteúdo, peça para que utilizem a ferramenta "FigJam Board" do Figma para elaborar os diagramas dos processos exemplificados nas atividades anteriores (Ir à mercearia comprar um refrigerante, fazer um suco de laranja e fazer pipoca na panela).

Proposta de solução:

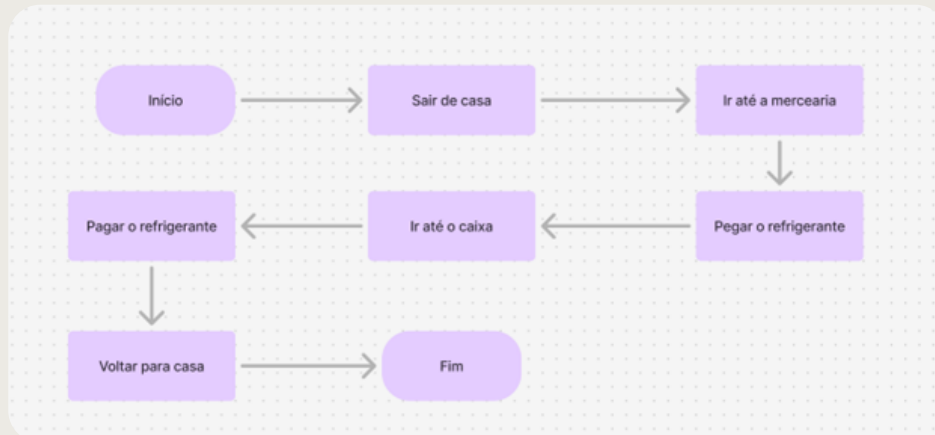
"Ir à mercearia comprar um refrigerante"

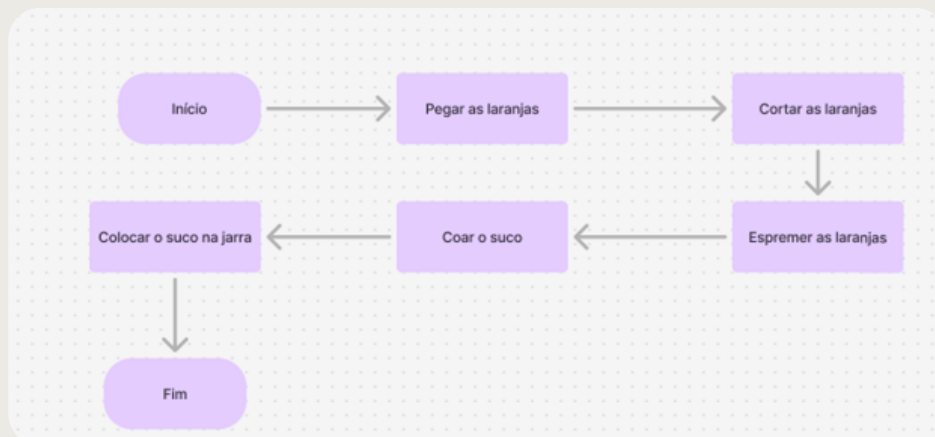
1. Após criar um novo arquivo no Figma, deverá utilizar as funções já prontas disponíveis. Inicie selecionando o "Rounded rectangle" e o desenhe na tela. Faça o mesmo com todos os símbolos necessários para montagem desse fluxograma. Nessa etapa você pode alterar a cor de cada símbolo se quiser.

Exemplo:

Depois de desenhar todos os símbolos, o resultado esperado é:

Após finalizar toda a estrutura do fluxograma, você deve adicionar o texto. As figuras já vêm com essa função, então não será preciso adicionar caixa de texto; é preciso apenas digitar o passo a ser executado em cada etapa.

Resultado:

“Fazer um suco de laranja”

Resultado:

“Fazer pipoca na panela”

Resultado:

AULA 02

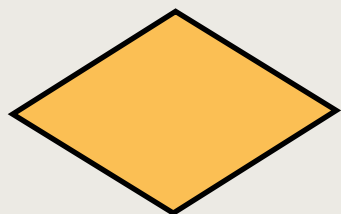
Resolução de problemas utilizando no fFluxograma o comando de decisão

Materiais necessários:
Computador, projetor e internet.

Duração:
1 hora e 30 minutos

Objetivo:
Apresentar e exemplificar o uso do símbolo de decisão ao diagramar um fluxograma.

ENGAJAR

Uma parte importante na criação de diagramas é a figura da "decisão". Representada por um losango, essa figura é essencial porque mostra escolhas ou caminhos diferentes em um processo, indicando como o sistema pode seguir por diferentes direções com base em certas condições que geralmente resultam em duas saídas: "falso" ou "verdadeiro". Vamos explorar a importância da figura de decisão nos fluxogramas, destacando sua função e como ela ajuda a deixar mais claro e eficiente o entendimento de procedimentos e sistemas complexos.

INVESTIGAR

Vamos realizar um exemplo para ajudar os alunos a entenderem melhor o assunto. A atividade é: **Ir nadar no rio, caso esteja chovendo você deverá ficar em casa.**

Proposta de solução:

Primeiro devemos fazer o algoritmo:

1. Consultar a previsão do tempo;
2. Está chovendo?

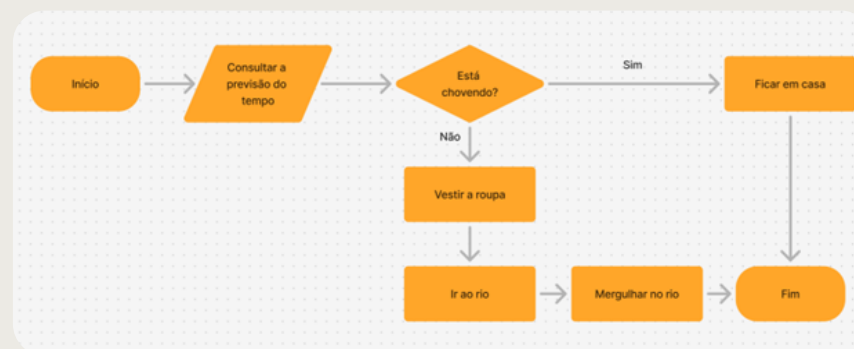
(SIM) 3. Ficar em casa;

(NÃO) 3. Vestir a roupa;

4. Ir ao rio;
5. Mergulhar no rio.

ATENÇÃO!

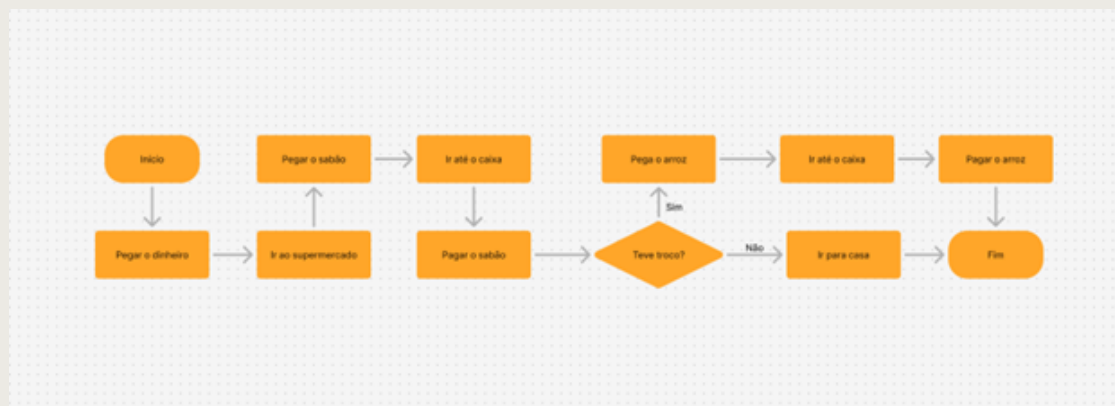
Um tópico importante que você deverá falar para seus alunos é que em toda condicional deve-se realizar uma pergunta. A depender da resposta o algoritmo seguirá um caminho específico. A pergunta que deverá ser feita na condicional não poderá ser ambígua, em outras palavras, o fluxo seguirá somente um caminho por vez.

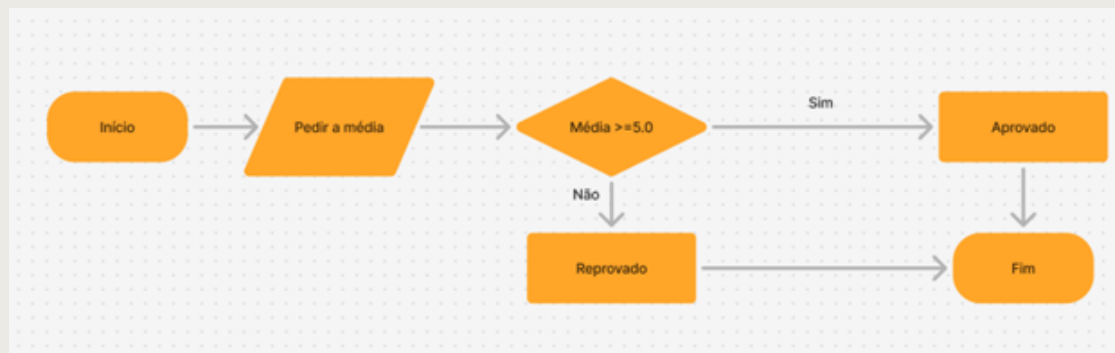
Agora devemos colocar as informações do algoritmo no fluxograma utilizando o Figma.

AGIR

Para que os alunos compreendam melhor o conteúdo, proponha que elaborem os fluxogramas para as seguintes atividades:

- Ir no mercado comprar sabão, se tiver troco, comprar arroz;
- Informar se o aluno passou de ano ou não, a média escolar é 5.

"Ir no mercado comprar sabão, se tiver troco, comprar arroz"

Proposta de solução:

"Informar se o aluno passou de ano ou não, a média escolar é 5"

Proposta de solução:

AULA 03

Resolução de problemas Utilizando nofluxograma o comando repetição

Materiais necessários:
Notebook, internet e projetor.

Duração:
1 hora e 30 minutos

Objetivo:
Apresentar os comandos de repetição e exemplificar seu uso, e aplicar ao Figma com suas funcionalidades.

ENGAJAR

1. CONTEXTUALIZAÇÃO/ENGAJAR

Nesta aula o(a) reaplicador(a) deverá ensinar os comandos de repetição com fluxograma.

Muitas são as situações em que temos que realizar tarefas repetitivas e para nós, humanos, fazer essas atividades que precisam de muita repetição torna-se monótono e chato. Já os computadores podem ser programados para realizar atividades repetitivas de uma forma muito mais rápida. Portanto, aprender a usar as estruturas de repetição é uma das coisas mais importantes no aprendizado de algoritmos.

INVESTIGAR

Nesse momento o(a) o repaplicado deverá mostrar o seguinte fluxograma para os alunos e pedir para que eles identificarem onde está o elemento de repetição no fluxograma.

COMENDO BISCOITOS

INÍCIO

PEGAR O PACOTE DE BISCOITO

ABRIR O PACOTE

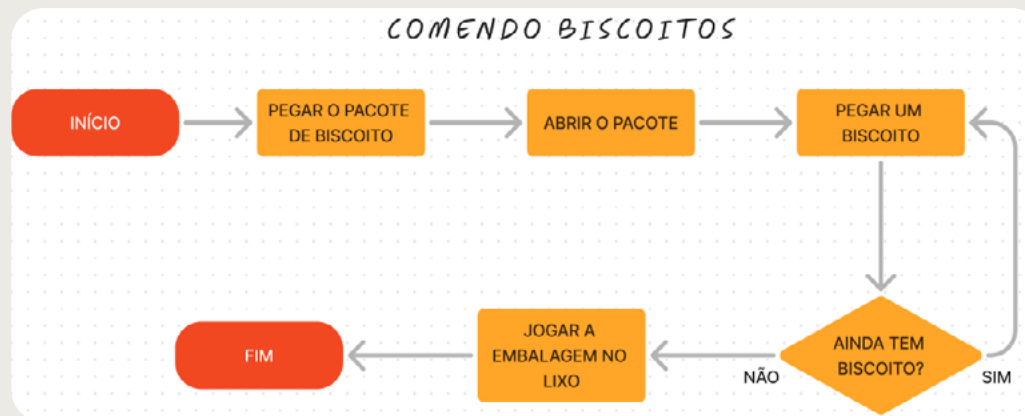
PEGAR UM BISCOITO

AINDA TEM BISCOITO?

SIM

NÃO

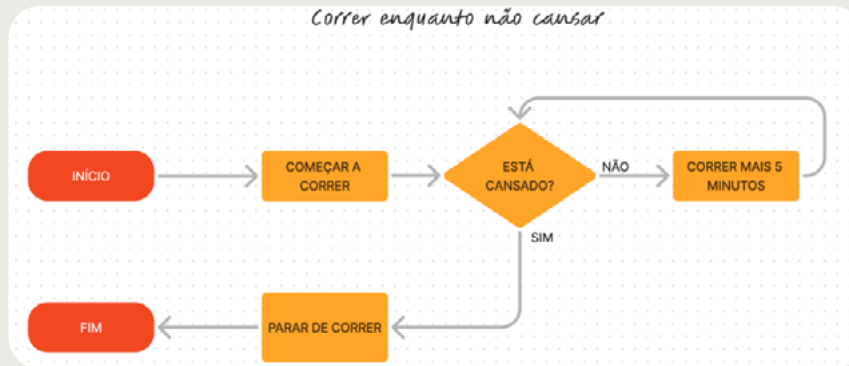
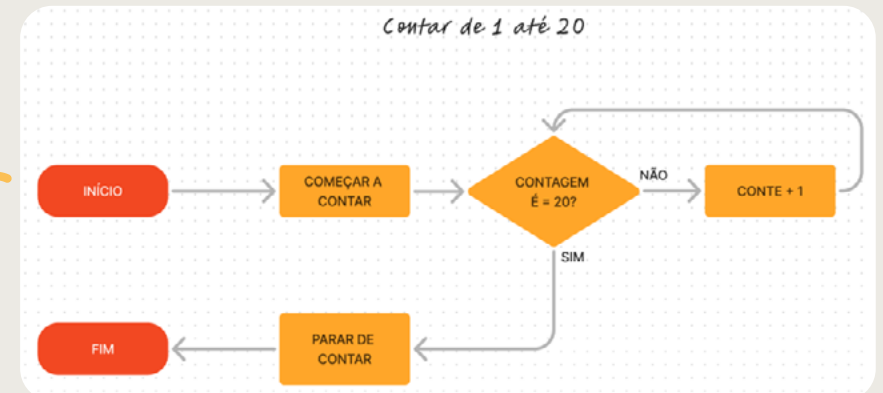
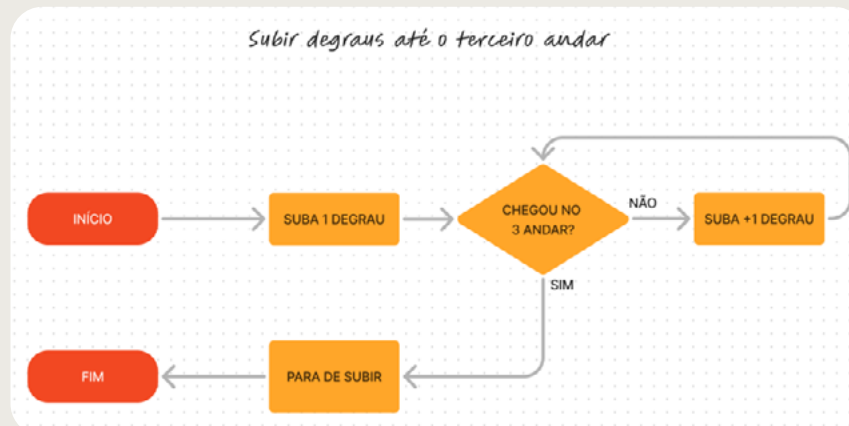
JOGAR A EMBALAGEM NO LIXO

FIM

A estrutura é bem parecida com a condição. A principal diferença entre a condicional e a repetição é que esta última quando possui na sua condição verdadeira um conjunto de instruções, podem ser feitas várias vezes repetitivamente.

Não se esqueça de falar para seus alunos que a estrutura de repetição não 'sabe' quantas vezes ocorreram repetições. Você pode até falar o seguinte exemplo: Cada pacote de biscoito pode vir com quantidades diferentes de biscoitos, então esse fluxograma vai servir para quaisquer tipos de biscoitos porque a instrução "Pegar um biscoito" vai se repetir quantas vezes for necessário.

AGIR

Peça para seus alunos analisarem as seguintes situações e desenhem o fluxograma das possíveis soluções utilizando a estrutura de repetição:

1. CONTINUE CORRENDO ENQUANTO NÃO CANSAR

2. CONTE DE 1 A 20

3. SUBA OS DEGRAUS DE UMA ESCADA ATÉ CHEGAR NO 3° ANDAR

AULA 04

Apresentação do Scratch

Materiais necessários:
Notebook, internet e projetor.

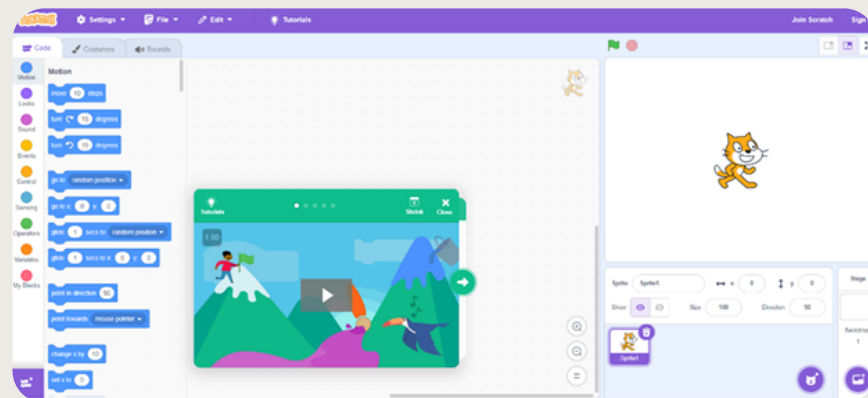
Duração:
1 hora e 30 minutos

Objetivo:
Apresentar a platafoma Scratch e apresentar variáveis, inputs e outputs.

ENGAJAR

Nesta aula o(a) reaplicador(a) deverá explicar brevemente o que é a plataforma Scratch, mostrando seus elementos principais. Além disso, os participantes deverão aprender o que são variáveis, inputs e outputs.

O Scratch é uma linguagem de programação visual projetada para ser fácil e acessível, especialmente para iniciantes, como crianças e adolescentes. Com o Scratch, os usuários podem criar animações, jogos e histórias interativas por meio de blocos de código que se encaixam como peças de um quebra-cabeça, sem a necessidade de escrever código tradicional.

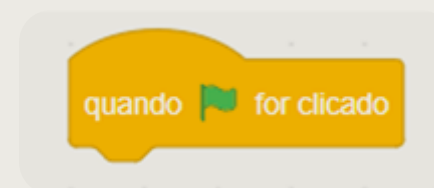
PRINCIPAIS ELEMENTOS

Palco:

O lugar onde o "gato" está se chama palco e é nele onde será possível ver os comandos que serão programados.

Comandos:

Na esquerda estão os comandos que usaremos para programar as ações do gato. Os comandos são separados por categorias e cada categoria tem uma cor. Para utilizá-las é só clicar e arrastar para a área localizada no meio, que será o espaço onde os comandos serão colocados.

É importante sinalizar que sempre iniciaremos os códigos utilizando os blocks de eventos, sinalizados pela cor amarela.

quando for clicado

Variáveis:

As variáveis são elementos que armazenam informações que podem ser usadas e modificadas ao longo do projeto. Elas permitem que os programadores guardem valores temporários, como números ou textos, para serem reutilizados em diferentes partes do código.

Input:

Refere-se aos dados ou informações que o programa recebe do usuário. No Scratch, inputs podem ser fornecidos por meio de cliques, teclas do teclado ou até mesmo perguntas (usando o bloco "pergunte e espere"). Esses inputs podem ser armazenados em variáveis para serem usados posteriormente no programa.

Output:

Refere-se aos dados ou resultados que o programa gera e apresenta ao usuário. No Scratch, outputs podem ser exibidos na tela por meio de diálogos, alterações visuais ou pela exibição do valor de uma variável (como a pontuação de um jogo, por exemplo).

INVESTIGAR

Mostre aos alunos como fazer o gato falar "Oi" usando o comando "Diga", localizado na aba de aparência.

Apresente o comando "diga" que pode ser encontrado em seção aparência, é esse comando que vamos usar para o gatinho falar.

diga oi por 2 segundos

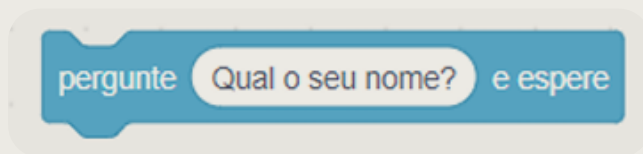
Exemplo prático com variáveis:

Peça para os alunos descobrirem como fazer o gato "Perguntar o nome do usuário" usando o comando "Pergunte". Ensine que a resposta será armazenada na variável "Resposta".

Resposta:

O comando correto é o "Pergunte".

O Scratch permite que sejam lidas as informações fornecidas pelo aluno que está programando. Primeiro você deverá ensinar aos seus alunos a entrada de informações. No Scratch, existe um bloco de comando que serve para pegar a entrada de informações, esse comando é o:

pergunte Qual o seu nome? e espere

Atividade 1

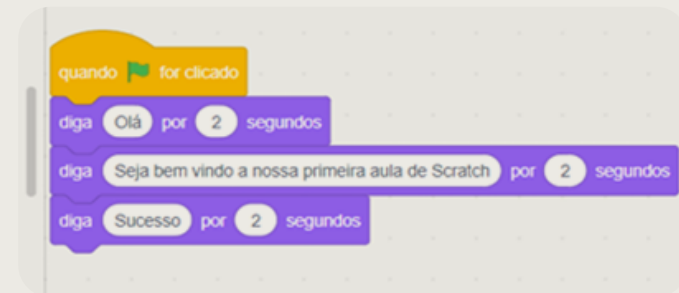
O problema agora é fazer o gatinho falar as seguintes frases, uma por vez:

1. Olá!

2. Seja bem-vindo a nossa primeira aula de Scratch.

3. Sucesso!

Diga a eles que os blocos podem ser usados quantas vezes forem necessários. Deixe que eles pensem um pouco. A resposta:

quando 🏳 for clicado
diga Olá por 2 segundos
diga Seja bem vindo a nossa primeira aula de Scratch por 2 segundos
diga Sucesso por 2 segundos

Atividade 2:

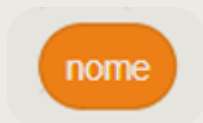
Desafio com variáveis e concatenação:

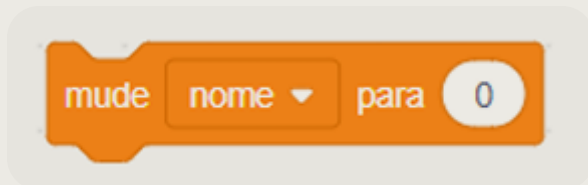
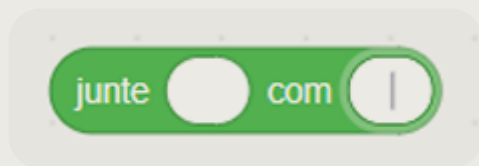
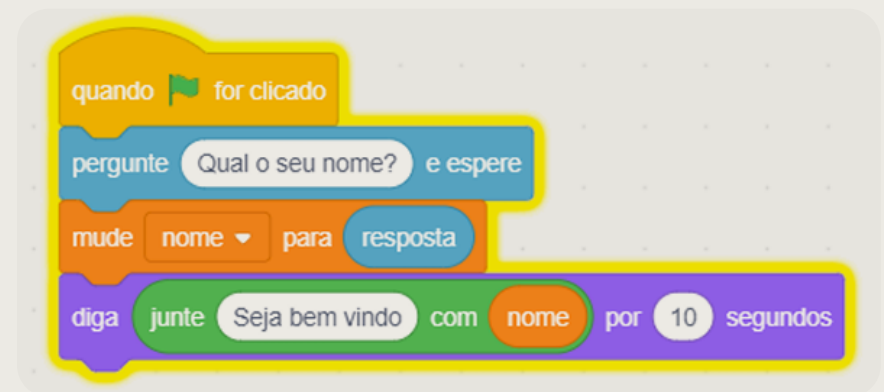
Agora, introduza o conceito de concatenação (juntar duas informações) para os alunos.

AGIR

O desafio é dizer uma frase mais o nome da pessoa; para isso, vamos precisar criar uma variável.

1- Criar uma variável chamada “nome”.

nome

2- Insira um valor para a variável usando o comando “mude”.

mude nome para 0

3- Insira o operador de “Concatenação” para juntar informações.

junte com

A resposta:

quando for clicado
pergunte Qual o seu nome? e espere
mude nome para resposta
diga junte Seja bem vindo com nome por 10 segundos

Seja bem vindo mirelly

AULA 04

Operadores

Materiais necessários:
Notebook e projetor.

Duração:
2 horas

Objetivo:
Apresentar os operadores matemáticos, de comparação e lógicos do Scratch. Mostrar como os operadores são usados para realizar cálculos e tomar decisões em um programa. Criar projetos simples para aplicar operadores.

ENGAJAR

Os operadores são elementos fundamentais tanto na matemática quanto na programação, desempenhando papéis cruciais em realizar cálculos e manipular valores.

Na matemática e na programação, os **"Operadores Aritméticos"** são usados para realizar operações básicas com números (adição (+), subtração (-), multiplicação (*) e divisão (/). Temos também os **"Operadores de Comparação"** que são utilizados para comparar dois valores, como por exemplo: o símbolo de "igual a" (==), "diferente de" (!=), "maior que" (>), "menor que" (<), "maior ou igual a" (>=) e "menor ou igual a" (<=).

Além disso, temos os **"operadores lógicos"** E, OU, NÃO.

Operadores desempenham um papel crucial na expressão de lógica e cálculos, tornando-se ferramentas essenciais tanto na resolução de problemas matemáticos quanto na implementação de algoritmos em programação.

INVESTIGAR

Vamos fazer um exercício. O problema é realizar a soma de dois números na calculadora, mas sem a calculadora.

Atividade

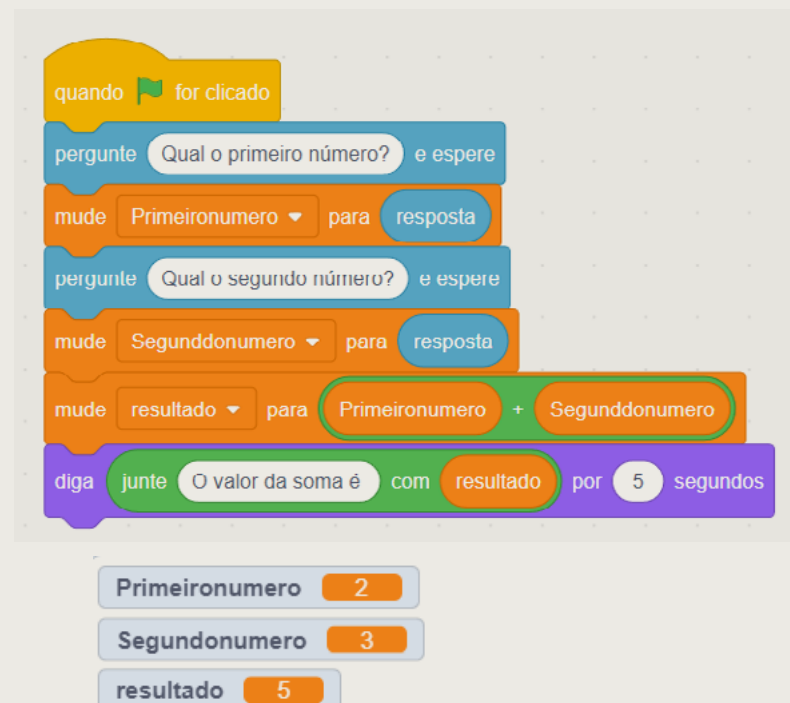
- Para a realização de um cálculo precisamos saber os números.

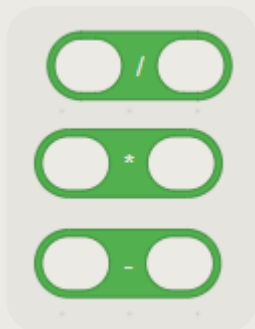
- Em seguida, digitar o primeiro número, usar o sinal de soma (+), digitar o segundo número e esperar a resposta.

Pergunte para os alunos o número mínimo de variáveis que seriam necessárias para realizar essa operação.

Resposta:

3 variáveis. Número 1, número 2 e resultado entre a operação dos dois primeiros números.

AGIR

Vamos fazer mais um:

Agora você deverá fazer os cálculos usando cada um dos operadores matemáticos: subtrair (-), multiplicar (*) e dividir (/).

ATIVIDADE COMPLEMENTAR

Desafie os alunos a fazer a seguinte atividade.

Calculadora de Troco:

Faça um código que calcule o valor de um troco ao informar o preço de dois itens de compra e um orçamento inicial.

Por exemplo, comprar banana e maçã com R$10,00.

AULA 04

Operadores

Materiais necessários:
Notebook e projetor.

Duração:
2 horas

Objetivo:
Iniciar a introdução a programação usando o programa Scratch.

ENGAJAR

No Scratch, os comandos de condição são usados para criar ramificações no código, permitindo que certos blocos de código sejam executados apenas se uma condição específica for atendida. Essas estruturas condicionais são fundamentais para criar lógica e tomadas de decisão em seus projetos. Aqui estão alguns dos principais comandos de condição no Scratch:

1. Bloco "Se" (`if`):
O bloco "Se" é usado para testar uma condição e executar um bloco de código se essa condição for verdadeira.

2. Bloco "Se-Senão" (`if-else`):
O bloco "Se-Senão" permite executar um bloco de código se a condição for verdadeira e outro bloco se a condição for falsa.

3. Bloco "Esperar Até que" (`wait until`):
Este bloco faz com que o programa aguarde até que uma condição específica seja verdadeira antes de continuar.

4. Bloco "Repita Até que" (`repeat until`):
Este bloco repete um bloco de código até que uma condição específica seja verdadeira.

5. Bloco "Esperar" (`wait`):
O bloco "Esperar" faz com que o programa aguarde por um determinado número de segundos antes de continuar.

Estes comandos de condição permitem que você crie lógica de controle de fluxo em seus projetos Scratch, tornando seus programas mais interativos e dinâmicos.

INVESTIGAR

Peça para que os alunos pensem em como eles resolveriam essa situação:

O Scratch precisa informar, perguntar a idade do usuário e caso a pessoa seja maior de idade ele responda: "Você é maior de idade".

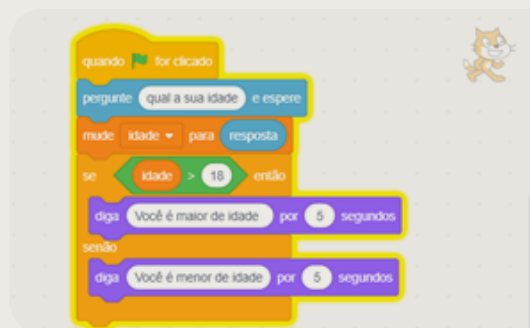
Pergunte quantas variáveis teríamos, e quais operadores usaremos.

Resposta para o desafio:

A
G
I
R

Agora que os alunos já têm uma noção sobre o uso de operadores lógicos e condicionais, peça para que eles alterem o código para quando a resposta for menor que 18, o Gato responda: "Você é menor de idade".

Para isso, eles devem utilizar o se/senão.

A resposta será:

Desafio 2

Faça o Gato perguntar dois números e dizer qual dos dois é maior.

Resposta:

AULA 04

Operadores

Materiais necessários:
Notebook e projetor.

Duração:
2 horas

Objetivo:
Apresentar e praticar as estruturas de repetição repetir e forever.

ENGAJAR

Laço de repetição

Assim como mostrado no fluxograma, no Scratch, o bloco de repetição é uma ferramenta essencial para executar um conjunto de comandos várias vezes sem a necessidade de repetir o código manualmente. Existem diferentes blocos de repetição disponíveis, cada um com um propósito específico. Aqui estão alguns exemplos de blocos de repetição no Scratch:

1.1. "Repetir" (`repeat`):

Este bloco executa um conjunto de comandos um número específico de vezes. Você define a quantidade de repetições.

1.2. "Repetir Até" (`repeat until`):

Este bloco repete um conjunto de comandos até que uma condição específica seja verdadeira.

1.3. "Para Sempre" (`forever`):

O bloco "Para Sempre" cria um loop infinito, executando continuamente o bloco de código dentro dele.

Os blocos de repetição são poderosos para simplificar a lógica de seus projetos no Scratch. Eles ajudam a evitar repetições desnecessárias de código, tornando seus scripts mais limpos e fáceis de entender.

2. Exemplos de Repita

2.1. Exemplo 1:

Faça o código usando o operador, diga 10 vezes e pergunte para os alunos como as operações apresentadas na aula poderiam ser utilizadas para solucionar essa atividade com menos linhas de código.

2.1. Exemplo 2:

Faça o código onde o personagem ande 50 passos para frente e diga: "Eu estou caminhando" repetidamente, 5 vezes.

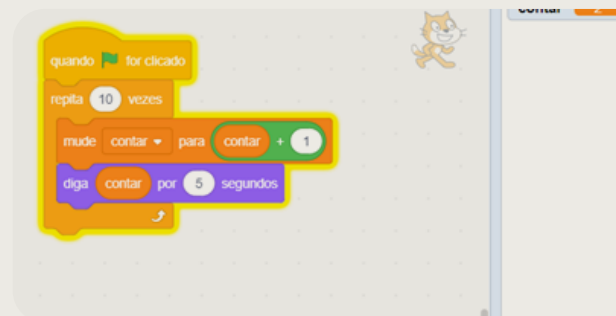
quando 🏁 for clicado
repita 5 vezes
mova 50 passos
espere 2 seg
diga Eu estou caminhando! por 2 segundos

2.1. Exemplo 3:

Crie um código onde o personagem caminhe para a direita até encostar na borda da tela. Utilize o bloco "Repetir Até" (repeat until) e a condição de que ele parará de andar quando tocar na borda.

quando 🏁 for clicado
repita até que tocando em borda ?
mova 10 passos
diga Eu estou correndo!
se tocar na borda, volte
defina o estilo de rotação para esquerda-direita

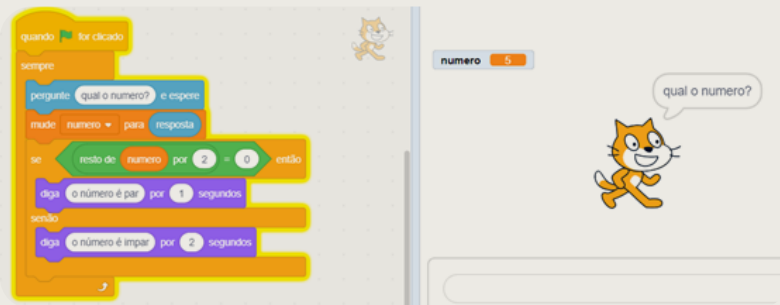
quando 🏁 for clicado
repita 10 vezes
mude contar para contar + 1
diga contar por 5 segundos

2

3. Exemplos de Forever

3.1. Exemplo 1:

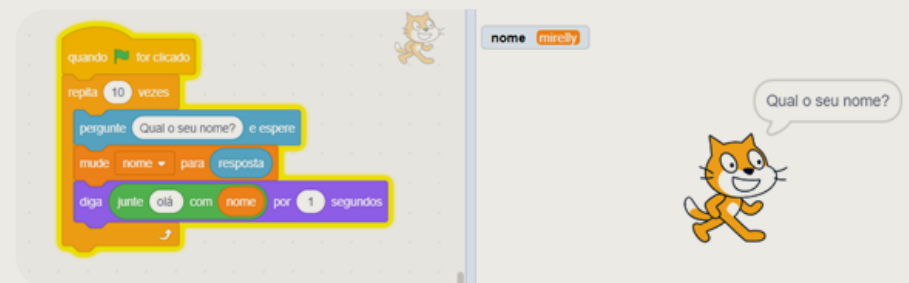
Os alunos deverão fazer um programa no Scratch para saber se um número informado é ímpar ou par utilizando o "Sempre". É preciso informar um número e verificar o resto de sua divisão

3.1. Exemplo 2:

Crie um projeto em que o personagem mude sua cor constantemente enquanto toca uma música de fundo. O personagem deve continuar mudando a cor indefinidamente enquanto a música toca. Utilize o bloco "Para Sempre" (forever).

4. DESAFIOS

4.1. Desafio 1:

Vamos realizar um desafio. O gato deverá solicitar o nome de 10 alunos e a cada pergunta ele deverá pronunciar o nome que foi digitado. Solicite aos alunos que pensem em uma resposta para esse desafio.

Resposta:

4.2. Desafio 2:

Crie um projeto onde um personagem fique pulando indefinidamente enquanto muda de tamanho. O personagem deve crescer 10 unidades, depois diminuir 10 unidades, e repetir essa ação continuamente. Utilize o bloco "Para Sempre" (forever) para que o personagem continue pulando e mudando de tamanho sem parar.

Resposta:

DESENVOLVIMENTO
WEB

COORDENADOR Marcos Vinicius Tavares Souza

AUTORES Tabatha Costa Bastos
Juliana da Silva Firmo
Tobias da Rodrigues Silva
Isaque Rodrigues Rocha
Vitória Sales de Jesus
Kawan Lucas Silva dos Santos
Mirelly Araújo Dos Santos
Renata Santos Celestino
Josué Pereira Da Crus
Marcos Vinicius Tavares Souza

REVISÃO Mariana Carolina de Almeida Souza

PROJETO GRÁFICO Patricia Karin S. Faber

MÓDULO 1

HTML/CSS/JS

Neste módulo, os participantes serão introduzidos ao universo do desenvolvimento front-end, explorando as bases do HTML e CSS, além de serem apresentados a uma nova linguagem de programação: o JavaScript (JS).

Ao final do módulo, espera-se que os alunos compreendam as principais ferramentas utilizadas na criação de aplicações e páginas web, além de terem familiaridade com as funcionalidades básicas da linguagem JavaScript (JS).

AULA 01

Introdução ao VS Code e ao HTML

Materiais necessários:
Computador, projetor e internet.

Duração:
1 hora e 30 minutos

Objetivo:
Conhecer a plataforma VS Code, explicar o que é o HTML e sua estrutura básica e apresentar as tag's h1 e p.

ENGAJAR

Comece a aula interagindo com os alunos perguntando para eles se já pensaram em como os sites são criados?

Após a reflexão inicial, explique de forma simples como sites podem ser criados utilizando o HTML.

"HTML é usado para criar a estrutura de um site, ou seja, definir os títulos, parágrafos, imagens, links e muito mais."

Após isso, apresente que é preciso utilizar um editor de código para utilizar o HTML e explique então o que é o VS Code.

"O Visual Studio Code, ou VS Code, é um editor de código muito usado para criar programas e páginas da web. Ele nos ajuda a escrever códigos de forma organizada e eficiente."

INVESTIGAÇÃO

Estrutura Básica

Nesse momento peça para que os alunos abram o VS Code e te acompanhem durante as explicações.

- Mostre como criar uma pasta no VS Code;
- Peça para que os alunos renomeiem a pasta com seus respectivos nomes;
- Mostre como criar um arquivo;
- Escreva no arquivo criado a estrutura básica do HTML e explique a importância de cada linha.

```
<!DOCTYPE html> "Informa ao navegador que o arquivo é um documento HTML."
<html> "A raiz do documento HTML"
<head> "Onde colocamos informações sobre a página (como o título)"
    <meta charset="UTF-8"> "Aceita caracteres especiais"
    <title>Document</title> "Define o nome que aparece na aba do navegador"
</head>
<body> "Contém todo o conteúdo visível da página"

</body>
</html>
```

Tags

Nesse momento, explique aos alunos o que são tags no HTML. Mostre como elas funcionam como “marcadores” que indicam o início e o fim de um elemento na página. Em seguida, peça que identifiquem onde as tags “abrem” e “fecham” na estrutura básica do HTML apresentada.

Apresente aos alunos as tags <h1> e <p>, explicando sua aplicação e importância, e faça um exemplo simples com os alunos.

```
<!DOCTYPE html> “Informa ao navegador que o arquivo é um documento HTML.”
<html> “A raiz do documento HTML”
<head> “Onde colocamos informações sobre a página (como o título)”
    <meta charset=”UTF-8”> “Aceita caracteres especiais”
    <title>Document</title> “Define o nome que aparece na aba do navegador”
</head>
<body> “Contém todo o conteúdo visível da página”
    <h1> Minha Primeira Aula de HTML </h2>
    <p> Minha Primeira Aula de HTML </p>
</body>
</html>
```

AGIR

Neste momento, peça para os alunos pesquisarem sobre algo que eles gostem de um tema previamente definido (o tema pode ser sobre esporte, comida, personagem de desenho, filme, série etc.). Mas é importante que seja escolhido o mesmo tema para todos os alunos.

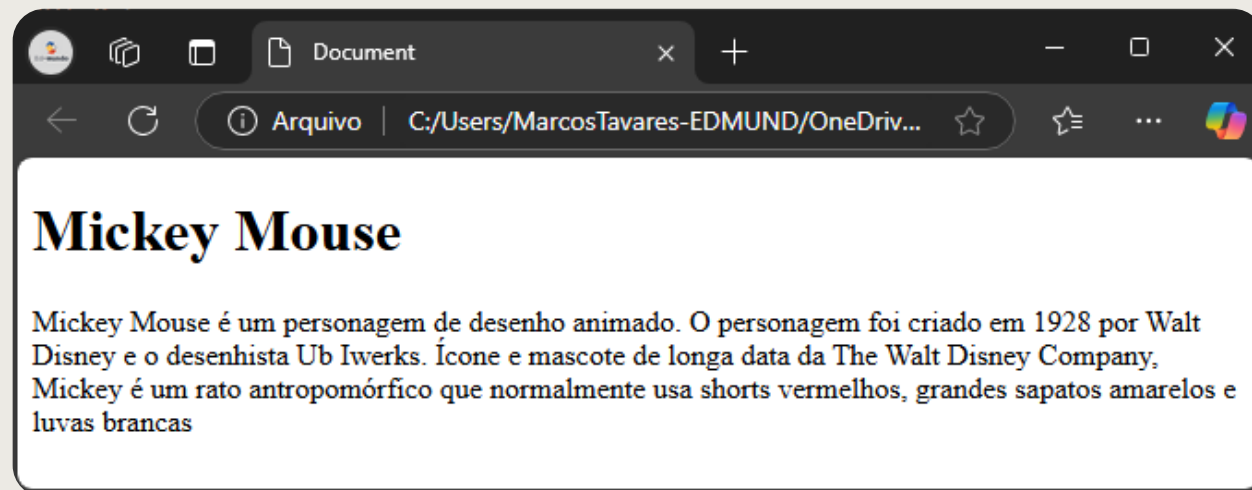
Nesse momento, peça para que os alunos escrevam um título com o nome da busca deles e, logo em seguida escrevam um parágrafo sobre o tema.

Exemplo:

Document

Arquivo | C:/Users/MarcosTavares-EDMUND/OneDriv...

Mickey Mouse

Mickey Mouse é um personagem de desenho animado. O personagem foi criado em 1928 por Walt Disney e o desenhista Ub Iwerks. Ícone e mascote de longa data da The Walt Disney Company, Mickey é um rato antropomórfico que normalmente usa shorts vermelhos, grandes sapatos amarelos e luvas brancas

AULA 02

Tag's "img", "a"

Materiais necessários:

Computador, projetor e internet.

Duração:

1 hora e 30 minutos

Objetivo:

Conhecer novas funcionalidades no HTML como as tag's "img" e "a".

ENGAJAR

Utilizando o exemplo criado na aula passada, pergunte aos alunos o que seria legal que tivesse na página criada.

Após ouvir as sugestões, fale que seria muito interessante ter uma foto do personagem para que a gente pudesse ver de fato como ele é. Além disso, seria interessante que houvesse um link para que a pessoa que ficou interessada nesse desenho pudesse ler mais.

Por isso, explique a eles a importância dos links e dos recursos gráficos em páginas web.

INVESTIGAR

Nesse momento, dê alguns minutos para os alunos pesquisarem qual tag utilizamos para inserir imagem e links nas páginas web.

Após a pesquisa deles explique que as tags são <img> e <a> e fale um pouco sobre cada uma delas.

> ***As tags <a> e <img> desempenham papéis fundamentais na criação de páginas web interativas e visualmente atraentes. A tag <a> permite a navegação entre páginas e recursos online. Já a tag <img> enriquece as páginas com imagens, tornando o conteúdo mais dinâmico, envolvente e fácil de entender.***

Após isso, explique aos alunos os atributos de cada tag e como utilizá-las.

Tag <img>: Usada para adicionar imagens à página.

Atributo principal: src (especifica o caminho da imagem).

Atributo adicional: alt (texto alternativo exibido quando a imagem não carrega).

Tag <a>: Usada para criar links clicáveis.

Atributo principal: href (especifica o destino do link)

Explique de forma prática aos alunos como adicionar uma imagem e um link no exemplo da aula passada. Você pode seguir os seguintes passos.

Adicinando imagem

Pesquise no navegador uma imagem legal que remete ao tema e faça download e salve na pasta criada.

Após salvar a imagem, insira a tag <img> e indique o caminho da imagem na propriedade “src”.

Indique também um texto alternativo que abrirá caso a imagem não carregue, na propriedade “alt”.

```
<img src="/mickey.png" alt="imagem do Mickey">
```

Adicionando Link

Insira um novo parágrafo com a tag <p> e escreva um texto indicando um link para saber mais sobre o tema.

```
<p>Saiba mais sobre este desenho incrível na <a href="https://pt.wikipedia.org/wiki/Mickey_Mouse ">página da Wikipédia</a>.</p>
```

Mostre aos alunos como ficou o resultado.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="./style/style.css">
</head>
<body>

    <h1>Mickey Mouse</h1>
    <img src="./mickey.png" alt="">
     <p>Mickey Mouse é um personagem de desenho animado. O personagem foi criado em 1928 por Walt
Disney e o desenhista Ub Iwerks. Ícone e mascote de longa data da The Walt Disney Company, Mickey é um
rato antropomórfico que normalmente usa shorts vermelhos, grandes sapatos amarelos e luvas brancas</p>
      <p>Saiba mais sobre este desenho incrível na <a href="https://pt.wikipedia.org/wiki/Mickey_Mouse
">página da Wikipédia</a>.</p>
</body>
</html>
```

Mickey Mouse

Mickey Mouse é um personagem de desenho animado. O personagem foi criado em 1928 por Walt Disney e o desen
sapatos amarelos e luvas brancas

Saiba mais sobre este desenho incrível na página da Wikipédia.

AGIR

Após a explicação de como utilizar as tag's, peça para que os alunos façam o mesmo com o tema escolhido por eles.

AULA 03

CSS

Materiais necessários:
Computador, projetor e internet.

Duração:
2 horas (a depender da desenvoltura da turma a aula deve ser dividida em duas)

Objetivo:
Apresentar o que é o CSS e aplicar algumas propriedades. Explicar sobre id e class.

ENGAJAR

Inicie a aula perguntando o que poderia ser melhorado na página criada até o momento.

Após ouvir os alunos, explique que a página poderia ter elementos mais bem distribuídos na tela, além disso, poderíamos ter cores diferentes no nosso arquivo.

Explique que para isso, usaremos o CSS.

> ***O CSS (Cascading Style Sheets) é a ferramenta que transforma páginas HTML simples em páginas personalizadas e mais agradáveis. Com o CSS é possível alterar cores, fontes e tamanhos sendo possível criar layouts responsivos.***

Compare com a vida real: o HTML é como a estrutura de uma casa, enquanto o CSS é a decoração que a torna acolhedora.

INVESTIGAR

Peça para os alunos pesquisem qual a tag utilizamos para conectarmos nossa página HTML ao CSS.

Após a pesquisa, explique que utilizamos a tag <link> para conectar nosso arquivo CSS ao HTML. Essa tag possui duas propriedades principais:

1. rel: Geralmente recebe o valor "stylesheet" para indicar que estamos referenciando uma folha de estilo CSS.

2. rel: Especifica o caminho onde o arquivo CSS está localizado, permitindo que o navegador carregue o estilo e o aplique à página.

Após isso, peça para que os alunos criem novos arquivos, um arquivo com extensão ".html" com o nome "aulacss" e outro com extensão ".css" com o nome "style".

Peça também para que os alunos escrevam no arquivo HTML a seguinte estrutura:

```
<!DOCTYPE html>
<html lang="PT-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Aula de CSS</title>
  <link rel="stylesheet" href="./style.css">
</head>
<body>
  <h1>Nome</h1>
  <p>Ed-mundo</p>
  <h1>Idade</h2>
  <p>15 anos</p>
</body>
</html>
```

Agora explique para eles o que iremos fazer:

Mudar as cores de fundo do nosso título ("h1") para marrom

Mudar as cores de fundo do nosso "body" para preto

Nesse momento, mostre que para personalizar nosso CSS devemos referenciar as tags no nosso arquivo. Para isso, podemos usar a própria tag. Explique aos alunos como mudar a cor de fundo do título para marrom.

Código CSS

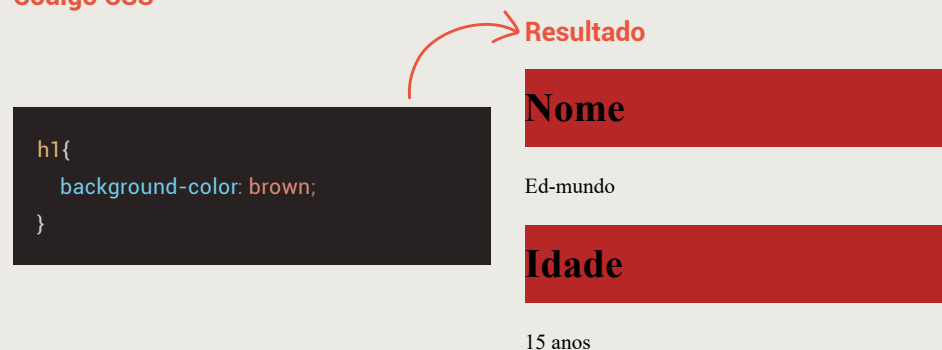
```
h1{
    background-color: brown;
}
```

Resultado

Nome

Ed-mundo

Idade

15 anos

Agora peça para que os alunos expliquem como é possível mudar a cor do "body" para preto.

Código CSS

```
h1{
    background-color: brown;
}
body{
    background-color: black;
}
```

Resultado

Nome

Idade

Questione aos alunos sobre o porquê das respostas do nome e da idade não aparecerem na tela e o que poderia ser feito para resolver esse problema.

Após a reflexão, explique que poderia ser mudados as cores de fundo da tag "p" ou poderia ser alterada a cor da tag "p" utilizando a propriedade "color" no CSS.

Mudando a cor da fonte da tag "p" para branco.

Código CSS

```
h1{
    background-color: brown;

}

body{
    background-color: black;
}

p{
    color: white;
}
```

Resultado

Nome

Ed-mundo

Idade

15 anos

Agora explique outra forma de referenciar seu código HTML no CSS seria utilizando as classes ou os ids.

Explique a eles que, se quiséssemos alterar a cor de fundo ou a fonte do nosso segundo título, não seria eficaz criar outra regra para o h1 no CSS. Isso acontece porque qualquer nova estilização aplicada à tag h1 substituiria a anterior, afetando todos os elementos h1 na página.

Por exemplo:

Código CSS

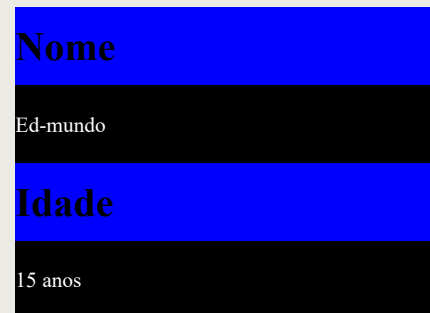
```
h1{
   background-color: brown;


}

body{
   background-color: black;
}

p{
   color: white;
}

h1{
   background-color: blue;


}
```

Resultado

Nome

Ed-mundo

Idade

15 anos

Por isso, é importante usar class e id, pois eles permitem personalizar elementos específicos do nosso conteúdo.

Para utilizar uma class, adicionamos o atributo class dentro da tag HTML e atribuímos a ele um nome representativo. No CSS, referenciamos essa class usando o caractere "." (ponto) seguido do nome definido.

Para utilizar um id, adicionamos o atributo id dentro da tag HTML e atribuímos a ele um nome exclusivo. No CSS, referenciamos esse id usando o caractere "#" (hashtag ou cerquilha) seguido do nome definido.

Explique de forma prática utilizando class como exemplo:

Código HTML

```
<!DOCTYPE html>
<html lang="PT-BR">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Aula de CSS</title>
   <link rel="stylesheet" href="./style.css">
</head>
<body>
   <h1>Nome</h1>
   <p>Ed-mundo</p>
   <h1 class="perguntadois">Idade</h2>
   <p>15 anos</p>
</body>
</html>
```

Código CSS

```
h1{
   background-color: brown;
}

body{
   background-color: black;
}

p{
   color: white;
}

.perguntadois{
   background-color: blue;
}
```

Resultado

Nome

Ed-mundo

Idade

15 anos

Explique, por fim, que a principal diferença entre class e id no CSS é que as classes são utilizadas para estilizar grupos de elementos que compartilham o mesmo estilo, ja os ID são usados para estilizar um elemento único e específico.

AGIR

Peça para que os alunos personalizem as páginas utilizadas nas aulas passadas, mudando a cor de fundo do body e mudando a cor da fonte do título e do parágrafo.

Exemplo

CSS

```
body{
    background-color: rgb(214, 14, 14);
}

p{
    color: azure;
}

h1{
    color: rgb(238, 241, 9);
}
```

Resultado

Mickey Mouse

Mickey Mouse é um personagem de desenho animado. O personagem foi criado em 1928 por Walt Disney e o desenhista Ub Iwerks. Ícone e mascote de longa data da The Walt Disney Company, Mickey é um rato antropomórfico que normalmente usa shorts vermelhos, grandes sapatos amarelos e luvas brancas

Saiba mais sobre este desenho incrível na página da Wikipédia.

AULA 04

Font-size, height, width, text-align

Materiais necessários:
Computador, projetor e internet.

Duração:
2 horas (a depender da desenvoltura da turma a aula deve ser dividida em duas)

Objetivo:
Continuar o estudo do CSS descobrindo novas propriedades como Font-size, height, width, text-align.

ENGAJAMENTO

Mostrando aos alunos os resultados que foram obtidos até o momento na nossa página, já temos uma página mais personalizada, porém alguns pontos poderiam ser melhores.

Pergunte aos alunos o que poderia ser melhorado na nossa página.

Após essa reflexão, fale que os textos estão desproporcionais com o tamanho da imagem. Além disso, poderia ser bacana se pudéssemos alinhar os elementos do site ao centro da página. Para isso, iremos ver nessa aula as propriedades Font-size, height, width, text-align .

INVESTIGAR

Primeiro, mostre aos alunos que assim como novo google docs, podemos personalizar tamanho e o estilo de nossas fontes. Para isso, é preciso utilizar as propriedades font-size e font-family.

FONT-SIZE

Para utilizar o font-size basta selecionar a tag de texto que você deseja alterar o tamanho e inserir o novo valor em "PX" (pixels) que você deseja.

Exemplo:

```
p{
   font-size: 40px;
}
```

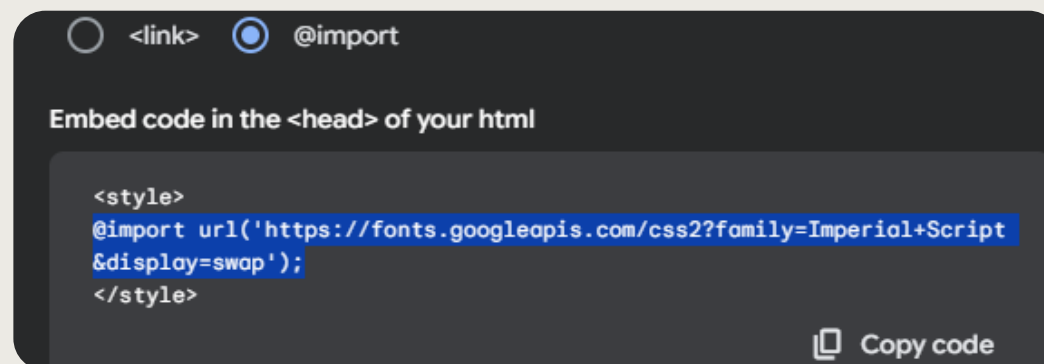
OBS :

Mostre ao aluno a diferença no tamanho da fonte.

FONT-FAMILY

Para mudar a fonte do texto podemos importar uma fonte através do site google fontes. Após pesquisar uma fonte que mais combine com seu conteúdo, clique em "get embed code"

Mude a opção para import e copie o código a partir do "@import" onde tem escrito "Embed code in the <head> of your html".

```
<style>
@import url('https://fonts.googleapis.com/css2?family=Imperial+Script&display=swap');
</style>
```

Código CSS:

```
@import
url('https://fonts.googleapis.com/css2?family=Imperial+Script&display=swap');
h1{
   color: rgb(238, 241, 9);
   font-family: "Imperial Script", cursive;
   font-size: 80px;
}
```

Mostre aos alunos o resultado da mudança.

HEIGHT E WIDTH

Explique aos alunos que height e width são, respectivamente, altura e largura em inglês. Explique que nesse caso podemos mudar as propriedades de altura e largura da imagem inserida apenas inserindo a propriedade e dando um valor em pixel, assim como na propriedade Font-size.

> **OBS:**
>
> Peça para que os alunos explorem esse recurso nas imagens do código deles.

TEXT-ALIGN

Por fim, mostre aos alunos que a propriedade text-align no CSS é utilizada para definir o alinhamento do texto ou do conteúdo dentro de um elemento HTML. Esse elemento recebe comumente as seguintes propriedades "left, right, center e justify;"

Após a explicação pergunte para os alunos: "Qual desses argumentos poderiam ser utilizados na seção body para centralizar todo o conteúdo?"

AGIR

Para o agir, peça para que os alunos apliquem os conhecimentos obtidos em aula e alterem os tipos e tamanho das fontes utilizadas em suas páginas. Os alunos devem também centralizar todo seu conteúdo na página e, por fim, ajustar o tamanho das imagens.

EXEMPLO:

Código CSS:

```css
@import
url('https://fonts.googleapis.com/css2?family=Imperial+Script&display=swap');
h1{
   color: rgb(238, 241, 9);
   font-family: "Imperial Script", cursive;
   font-size: 120px;
}
body{
   background-color: rgb(214, 14, 14);
   text-align: center;
}

p{
   color: white;
   font-size: 40px;
}

img{
   height: 450px;
}
```

Resultado

Mickey Mouse

Mickey Mouse é um personagem de desenho animado. O personagem foi criado em 1928 por Walt Disney e o desenhista Ub Iwerks. Ícone e mascote de longa data da The Walt Disney Company, Mickey é um rato antropomórfico que normalmente usa shorts vermelhos, grandes sapatos amarelos e luvas brancas

Saiba mais sobre este desenho incrível na página da Wikipédia.

AULA 05

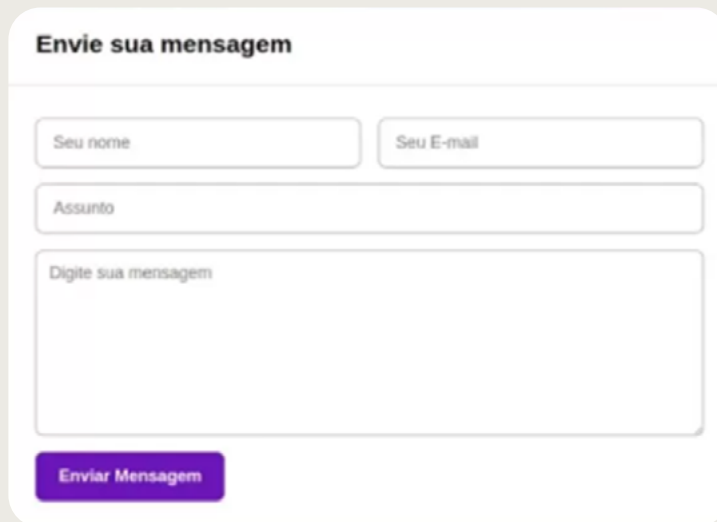
Formulários

Materiais necessários:

Computador, projetor e internet.

Duração:

1 hora e 30 minutos

Objetivo:

Ensinar os fundamentos da criação de formulários com HTML, abordando as principais tags e seus usos práticos.

ENGAJAR

Explique aos alunos o papel dos formulários em um site:

Um formulário em HTML é uma área onde o usuário pode inserir informações. Quando falamos em “informações”, pode ser qualquer coisa, como o nome, e-mail, idade, ou até uma mensagem. Essas informações são enviadas para um servidor para que sejam processadas e isso acontece quando clicamos no botão de enviar. Ele é a principal maneira de coletar informações de visitantes de um site, como contatos, inscrições e feedbacks.

Mostre um exemplo do formulário para os alunos:

Apresente cada uma das tags que iremos utilizar na aula:

<form> - Usada para definir o bloco do formulário e agrupar elementos.

Exemplo:

<form></form>

<label> - Serve para rotular elementos de formulário, como campos de texto, caixas de seleção e botões de opção. O label indica o que deve ser inserido em cada campo, facilitando a experiência do usuário.

Exemplo:

<label for="name">Nome:</label>

<input> - É um campo para que o usuário possa inserir algum texto, data, número, cor, etc. possui como principais atributos type, que recebe como valor o tipo do input. A tag em um formulário HTML é um elemento usado para criar campos de entrada interativos, nos quais as pessoas usuárias podem inserir dados.

Exemplo:

<input type="text"> <!-- Campo para texto simples -->

<input type="email"> <!-- Campo para e-mail -->

<textarea> - É um campo de entrada de texto que permite que os utilizadores digitem textos extensos.

Exemplo:

<textarea rows="4" cols="50"></textarea>

<button> - É um elemento que permite adicionar botões clicáveis a uma página web. Os botões podem ser usados em formulários ou em qualquer parte de uma página

Exemplo:

<button type="submit">Enviar</button>

INVESTIGAR

Peça para que os alunos façam uma pesquisa no google para explorar os elementos aprendidos. Eles podem pesquisar:

- Exemplos de formulários em sites que visitam.
- Diferença entre input type="text" e input type="email".

Construção de um formulário simples:

Construa um formulário simples junto com os alunos, explicando cada linha de código enquanto escreve, para fixar as tags ensinadas na aula:

Ex.:

```
<form>
  <label for="name">Nome:</label>
  <input type="text" id="name" name="name"><br><br>

  <label for="email">E-mail:</label>
  <input type="email" id="email" name="email"><br><br>

  <label for="message">Mensagem:</label><br>
  <textarea id="message" name="message" rows="4" cols="50"></
textarea><br><br>

  <button type="submit">Enviar</button>
</form>
```

Peça para que os alunos criem um formulário básico contendo:

- Nome (campo de texto)
- E-mail (campo de e-mail)
- Mensagem (área de texto)
- Botão de envio

```
<form>
  <label for="name">Nome:</label>
  <input type="text" id="name" name="name"><br><br>

  <label for="email">E-mail:</label>
  <input type="email" id="email" name="email"><br><br>

  <label for="message">Mensagem:</label><br>
  <textarea id="message" name="message" rows="4"
cols="50"></textarea><br><br>

  <button type="submit">Enviar</button>
</form>
```

AULA 06

Formulários

Materiais necessários:

Computador, projetor e internet.

Duração:

1 hora e 30 minutos

Objetivo:

Melhorando a usabilidade do formulário com validação e estilização com CSS.

ENGAJAR

Revise os elementos do formulário criados na aula anterior e destaque como podemos melhorar a usabilidade com validações (placeholder e required).

Após isso, introduza o assunto da aula de hoje:

Placeholder:

É um atributo que fornece ao usuário uma dica sobre o valor esperado em um campo de entrada. Ele é visível até que o usuário comece a digitar.

Exemplo:

```
<input type="text" placeholder="Digite seu nome">
```

Required:

O atributo “required” em HTML é um atributo booleano que indica que um campo de formulário é obrigatório para o envio do formulário.

Ao adicionar o atributo “required” a um campo de formulário, o navegador obriga o usuário a inserir dados naquele campo antes de enviar o formulário.

Exemplo:

```
<input type="email" required>
```

Agora apresente os conceitos de hover e padding que serão utilizados na nossa estilização do arquivo CSS:

Hover:

É uma pseudo-classe CSS que permite alterar o estilo de um elemento quando o usuário passa o mouse sobre ele. É uma técnica de design que serve para criar interatividade e melhorar a experiência do usuário.

O hover é muito utilizado em botões, logos e imagens. Por exemplo, é possível usar o hover para:

- Indicar que um link é clicável, alterando a cor de fundo, a cor do texto ou as fontes.
- Alterar a cor de um botão.
- Aumentar o tamanho de um ícone.

Exemplo:

```
button:hover {
  background-color: #4CAF50;
  color: white;
  }
```

Padding:

O padding em CSS é uma propriedade que define o espaçamento interno de um elemento, ou seja, a distância entre o conteúdo e a borda do elemento.

Exemplo:

```
input, textarea {
  padding: 10px;
}
```

INVESTIGAR

Adicione placeholder e required no formulário da aula anterior junto com os alunos e depois, aplique estilizações básicas com CSS.

Exemplo Html:

```
<form>

  <label for="name">Nome:</label>

  <input type="text" id="name" name="name" placeholder="Digite seu
nome" required><br><br>



  <label for="email">E-mail:</label>

  <input type="email" id="email" name="email" placeholder="Digite seu
e-mail" required><br><br>

  <label for="message">Mensagem:</label><br>

  <textarea id="message" name="message" placeholder="Digite sua
mensagem" rows="4" cols="50" required></textarea><br><br>

  <button type="submit">Enviar</button>

</form>
```

Exemplo CSS:

```
input, textarea {

  padding: 10px;

  width: 100%;

  box-sizing: border-box;

}

button:hover {

  background-color: #4CAF50;

  color: white;

}
```

AGIR

(30 min)

Agora que os alunos já praticaram e exploraram os conceitos de placeholder, required, hover e padding no "Investigar", eles criarão um formulário mais completo e estilizado do zero.

Proposta:

Crie um formulário de inscrição para um evento fictício, como um torneio de games, uma festa temática ou uma competição de talentos.

Requisitos:

- Inclua pelo menos 3 campos de entrada (input) com placeholder e required (ex.: Nome, E-mail, e Telefone).
- Adicione um botão com efeito hover (mude cor ou tamanho ao passar o mouse).
- Utilize padding nos campos para melhorar o espaçamento interno.
- Inclua uma área de texto (textarea) para comentários ou mensagens.

Passos para a Atividade:

Oriente os alunos a escrever o HTML do formulário, incluindo todos os elementos necessários.

Peça para aplicarem estilos básicos com CSS para deixar o formulário visualmente agradável.

Exemplos do arquivo Html do Formulário Finalizado:

```
<form>
  <label for="name">Nome:</label>
  <input type="text" id="name" placeholder="Digite seu nome" required><br><br>

  <label for="email">E-mail:</label>
  <input type="email" id="email" placeholder="Digite seu e-mail" required><br><br>

  <label for="message">Mensagem:</label>
  <textarea id="message" placeholder="Escreva sua mensagem aqui"></
textarea><br><br>

  <button type="submit">Enviar</button>
</form>
```

Exemplo do arquivo CSS:

```
input, textarea {
  padding: 10px;
  width: 300px;
  margin-bottom: 10px;
 }

 button {
 padding: 10px 20px;
 background-color: #4CAF50;
 color: white;
 border: none;
 cursor: pointer;
}

 button:hover {
 background-color: #45a049;
}
```

AULA 07

Listas

Materiais necessários:

Computador, projetor e internet.

Duração:

1 hora e 30 minutos

Objetivo:

Ensinar o uso de listas em HTML(UL, OL, LI) enquanto os alunos avaliam suas características pessoais e definem ideias de negócio conectadas a essas habilidades.

ENGAJAR

Nessa aula iremos aprender sobre listas. As listas no HTML são muito importantes para a organização e apresentação de informações textuais em uma página web.

 usamos essa tag para definir uma lista não ordenada.

 diferente da outra tag, essa daqui usamos para lista ordenada.

 essa tag tem a finalidade de listar cada item, iremos usá-lo de acordo com a quantidade de itens da nossa lista, sendo ela ordenada ou não ordenada.

Discussão inicial: Pergunte aos alunos: **"Quais características você acha que são importantes para quem quer empreender?"**

Exemplo visual: Mostre exemplos de listas de itens que aparecem em páginas de negócios, como sites de e-commerce (produtos listados) ou blogs (listas de dicas para empreendedores).

Explique que as listas em HTML (UL, OL, LI) ajudam a organizar ideias de forma clara e que, como empreendedores, organizar ideias e características pessoais pode ser essencial para o sucesso do negócio.

INVESTIGAR

Mostre como criar listas não ordenadas (UL) e ordenadas (OL) em HTML. Cada item da lista será uma característica pessoal que os alunos acreditam ser importante para empreender.

Ex.:

```
<ul>
  <li>Criatividade</li>
  <li>Resiliência</li>
  <li>Determinação</li>
</ul>
```

```
<ul>
  <li>Criatividade</li>
  <li>Resiliência</li>
  <li>Determinação</li>
</ul>
```

Peça para os alunos listarem características pessoais que podem influenciar o tipo de negócio que eles gostariam de criar. Utilize OL para listar as ideias de negócio (exemplo abaixo).

Ex.:

```
<ol>
  <li>Loja online de artesanato</li>
  <li>Aplicativo de saúde mental</li>
  <li>Serviço de consultoria financeira</li>
</ol>
```

```
<ol>
  <li>Loja online de artesanato</li>
  <li>Aplicativo de saúde mental</li>
  <li>Serviço de consultoria financeira</li>
</ol>
```

AGIR

Os alunos devem criar uma página HTML simples que inclua:

Uma lista UL com suas características pessoais principais.

Uma lista OL com sua ideia de negócio.

Ao final, os alunos devem justificar em uma breve descrição como suas características pessoais se alinham à ideia de negócio, destacando como isso pode influenciar o sucesso do projeto.

Ex.:

```
<!DOCTYPE html>

<html lang="pt-BR">

<head>

   <meta charset="UTF-8">

   <meta name="viewport" content="width=device-width, initial-scale=1.0">

   <title>Meu Negócio</title>

</head>

<body>

   <h1>Minhas Características Pessoais</h1>

   <ul>

         <li>Criatividade</li>

         <li>Resiliência</li>

         <li>Determinação</li>

   </ul>

     <h2>Minhas Ideias de Negócio</h2>

   <ol>

      <li>Loja online de artesanato</li>

   </ol>

     <p>Acredito que minha criatividade e resiliência me ajudam a pensar em ideias ino-
vadoras para o meu negócio e superar desafios durante o processo de construção. Além
disso, minha organização facilita a gestão das operações do negócio.</p>

</body>

</html>
```

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Meu Negócio</title>
</head>
<body>
   <h1>Minhas Características Pessoais</h1>
   <ul>
         <li>Criatividade</li>
         <li>Resiliência</li>
         <li>Determinação</li>
   </ul>

   <h2>Minhas Ideias de Negócio</h2>
   <ol>
      <li>Loja online de artesanato</li>
   </ol>

   <p>Acredito que minha criatividade e resiliência me ajudam a pensar em ideias inovadoras
para o meu negócio e superar desafios durante o processo de construção. Além disso, minha
organização facilita a gestão das operações do negócio.</p>
</body>
</html>
```

AULA 08

Menu flutuante (Parte 1)

Materiais necessários:
Computador, projetor e internet.

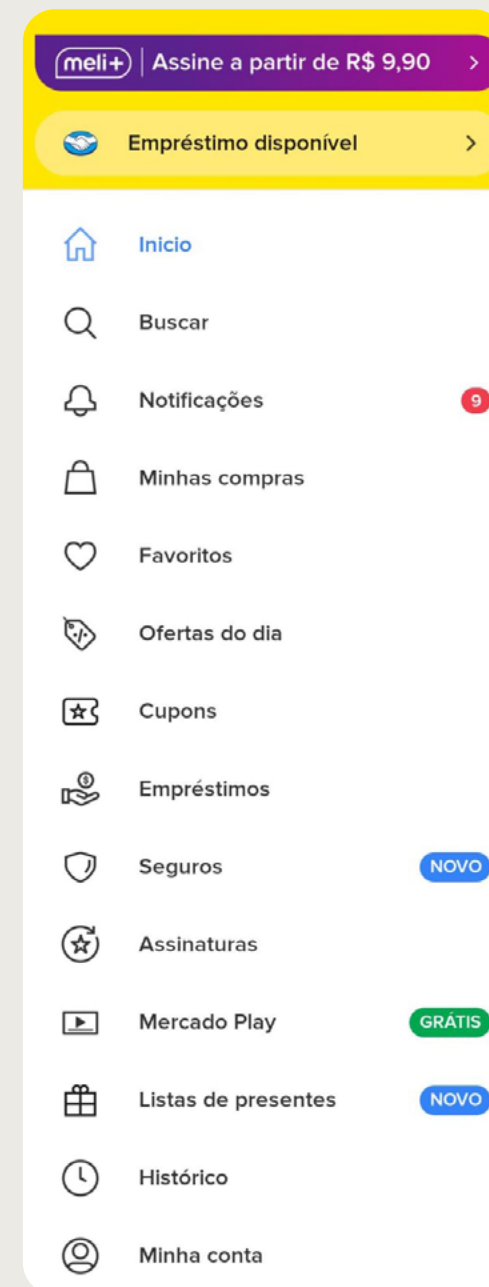
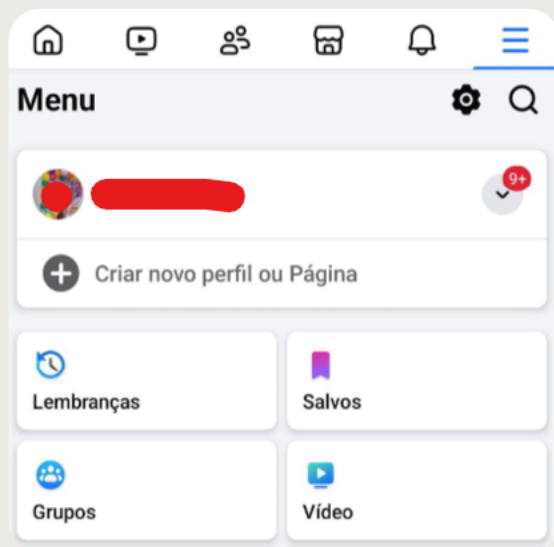
Duração:
1 hora e 30 minutos

Objetivo:
Iniciar o desenvolvimento de um menu flutuante, utilizando listas ordenadas e não ordenadas.

ENGAJAR

Explique para os alunos o que é um menu flutuante:

Um menu flutuante, também conhecido como menu fixo ou menu sticky, é um elemento de navegação que permanece visível na tela mesmo quando o usuário rola para baixo em uma página da web. Ele é geralmente posicionado no topo ou na lateral da página e oferece acesso rápido e fácil aos principais links e seções do site.

Mostre para os alunos exemplos de sites conhecidos que tenham menus flutuantes:

INVESTIGAR

Peça para que os alunos pesquisem como as listas podem ser aplicadas para construção do menu.

Resposta:

As listas podem ser aplicadas na construção de menus de navegação organizando os links de forma hierárquica e estruturada. Utilizando uma lista não ordenada (**<ul>**), os itens do menu são representados por elementos de lista (**<li>**), que contêm links (**<a>**). Essa estrutura permite agrupar e estilizar os itens do menu de forma simples com CSS, criando botões ou opções clicáveis.

Por exemplo:

Cada item de menu (como "Home", "Sobre", "Contato") pode ser um <**li**>, e o conjunto desses itens será organizado dentro de um <**ul**>.

CSS é utilizado para ocultar e exibir o menu, aplicar estilos, cores e efeitos de hover, transformando a lista básica em um menu visualmente atrativo.

Peça para que os alunos criem um arquivo chamado index.html e crie a estrutura do nosso menu flutuante junto com eles.

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Menu Flutuante</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div id="menu-icon">    </div>

  <div id="menu-container" class="hidden">
    <ul>
      <li><a href="#">Opção 1</a></li>
      <li><a href="#">Opção 2</a></li>
      <li><a href="#">Opção 3</a></li>
      <li><a href="#">Opção 4</a></li>
    </ul>
  </div>
</body>
</html>
```

Agora, o foco é na estilização e transições suaves.

Introduza o conceito de **transições CSS**, mostrando como isso melhora a experiência do usuário.

Explique também como **listas não ordenadas** podem ser estilizadas para se parecer com menus modernos.

Escreva o CSS com os alunos:

```
body {
   font-family: Arial, sans-serif;
}

/* Estilização do ícone do menu */
#menu-icon {
   font-size: 30px;
   cursor: pointer;  /* Altera o cursor para indicar que é clicável */
   position: relative;
   margin: 20px;
}

/* Estilização do contêiner do menu */
#menu-container {
   position: absolute;
   top: 50px;
   background-color: #fff;
   border: 1px solid #ccc;
   box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
   width: 200px;
   z-index: 1000;
}

/* O menu inicialmente fica escondido */
#menu-container.hidden {
   display: none;
}
```

```
/* Lista de itens no menu */
#menu-container ul {
   list-style-type: none;
   padding: 0;
   margin: 0;
}

/* Itens individuais do menu */
#menu-container ul li  {
   padding: 10px;
   margin: 1px solid #ccc;
}

/* Links do menu */
#menu-container ul li a {
   text-decoration: none;
   color: #333;
}

/* Estilização ao passar o mouse sobre os itens */
#menu-container ul li a:hover {
   background-color: #f0f0f0;
   transition: background-color 0.3s ease; /* Transição suave na cor de fundo */
}
```

AULA 08

Menu flutuante (Parte 2)

Materiais necessários:

Computador, projetor e internet.

Duração:

1 hora e 30 minutos

Objetivo:

Finalizar o desenvolvimento do menu flutuante com interatividade usando o JavaScript. Explorar funções, DOM e eventos de clique.

ENGAJAR

Rever o que foi feito na aula anterior:

Mostrar o estado atual do menu flutuante (HTML e CSS), que está estilizado, mas ainda sem interatividade.

Pergunte:

“Agora que nosso menu está estilizado, o que falta? Pensem em quando usamos menus em outros sites. O que acontece quando clicamos no ícone do menu?”

Explique para os alunos que isso é feito com a linguagem de programação JavaScript.

HTML
structure

CSS
presentation/appearance

JavaScript
dynamism/action

O JavaScript é uma das linguagens de programação mais usadas no mundo. É o que faz os sites se tornarem interativos. Usando uma analogia simples, o HTML é o esqueleto, CSS é a pele e as roupas, e o JavaScript é o cérebro que faz tudo funcionar. Se você clicar em algum botão da página ou ícone, e aparecer uma janela, isso é o Javascript fazendo a alteração do site ou aplicativo.

Como adicionar o JavaScript em um Arquivo HTML:

Você pode fazer isso usando a tag <script></script>, que deve abranger todo código JS que você escrever. O código JS pode ser adicionado: Entre as tags <head> Entre as tags <body>

Ex:

```
<html lang="pt-BR"></html>
<head>
  <meta charset="UTF-8">
  <title>Exemplo Simples</title>
  <script>
    alert("Hello world!"); // Exibe um alerta ao carregar a página
  </script>
</head>
<body>
  <h1>Bem-vindo!</h1>
</body>
</html>
```

INVESTIGAR

Peça aos alunos para pesquisarem sobre eventos de clique no JavaScript.

Oriente-os a buscar informações sobre o método addEventListener e como ele pode ser utilizado para capturar cliques em elementos.

Orientação: “Procurem exemplos de como capturar eventos de clique em botões ou ícones e como podemos alterar o comportamento de elementos com base nesses cliques. O que acontece quando clicamos e queremos alterar o estado de algo na página?”

AGIR

Peça para que os alunos criem um arquivo chamado script.js que será o arquivo onde deixaremos todo o código do javascript. Construa junto com os alunos o seguinte código:

```
// script.js
document.addEventListener('DOMContentLoaded', () => {
    const menuIcon = document.getElementById('menu-icon');
    const menuContainer = document.getElementById('menu-container');

   // Alternar para abrir e fechar o menu
    menuIcon.addEventListener('click', () => {
        menuContainer.classList.toggle('hidden');
    });

    // Fechar o menu ao clicar fora dele
    document.addEventListener('click', (event) => {
        if                    (!menuIcon.contains(event.target)                    &&
!menuContainer.contains(event.target)) {
            menuContainer.classList.add('hidden');
        }
    });
});
```

Explicação do Código:

DOMContentLoaded:
Assegura que o JavaScript só será executado após o carregamento completo da página.

addEventListener('click'):
Captura cliques no ícone do menu e alterna a visibilidade do menu usando a classe hidden.

document.addEventListener('click'):
Fecha o menu ao clicar fora dele, verificando se o clique foi no ícone ou no menu.

AULA 09

Calculadora (Parte 1)

Tema:

Tabelas

Materiais necessários:

Computador, projetor e internet.

Duração:

2 horas

Objetivo:

Apresentar as tags para criação de tabelas em HTML (Table, tr, td e th)

ENGAJAR

Inicie a aula fazendo as seguintes perguntas aos alunos:

“O que vocês entendem por tabela?”

“Onde vocês já viram tabelas sendo usadas?”

Após a discussão, apresente um exemplo de tabela para os alunos.

Nome	Idade		Altura
João	15		1.70m
Maria	16		1.65m
Ana	14		1.60m
Pedro	15		1.75m
Luiza	16		1.68m

Explique para eles que as tabelas são úteis para organizar dados de forma clara e estruturada e possuem linhas (horizontais) e colunas (verticais). Pergunte para os alunos quantas linhas e quantas colunas temos no exemplo mostrado.

No HTML as tabelas podem ser aplicadas não apenas para dados, mas também como base para layouts, como o de uma calculadora.

INVESTIGAR

Explique que uma tabela no HTML é criada usando as seguintes tags principais:

<table>: Define a tabela.

<tr>: Define uma linha da tabela (table row).

<td>: Define uma célula de dados (table data).

<th>: Define um cabeçalho da tabela (table header).

Mostre a seguinte imagem para os alunos e faça a pergunta a seguir:

“Imaginando essa calculadora como uma tabela, quantas linhas e quantas colunas teríamos?”

Após as respostas, confirme que apesar de termos as teclas “0” e de “=” ocupando mais espaços, a tabela está disposta em 6 linhas e 4 colunas.

Faça passo a passo de como inserir as 3 primeiras linhas da calculadora com os alunos. Explique que apesar de termos definido como uma tabela de 6 linhas, a primeira linha irá mostrar apenas as operações que a calculadora realizará, recurso que só irá ser feito posteriormente com ajuda do JavaScript.

Primeiramente insira a estrutura básica.

No “body” insira uma “div” com a class “fundo” para posteriormente editarmos o fundo da tela.

Dentro da primeira “div” insira um título “calculadora” e abra mais uma “div” com a classe “calculadora”.

Nessa nova div insira uma tag “p” com a id “resultado” e deixe a tag sem texto, pois será onde os resultados serão exibidos.

Finalmente, agora insira a tag <table> seguida de uma tag de linha <tr>;

Dentro da tag de linha insira as informações que aparecem na primeira linha com a tag <dt> e adicione a classe “botao” para cada uma das tags <dt>.

Repita o processo para as próximas 2 linhas.

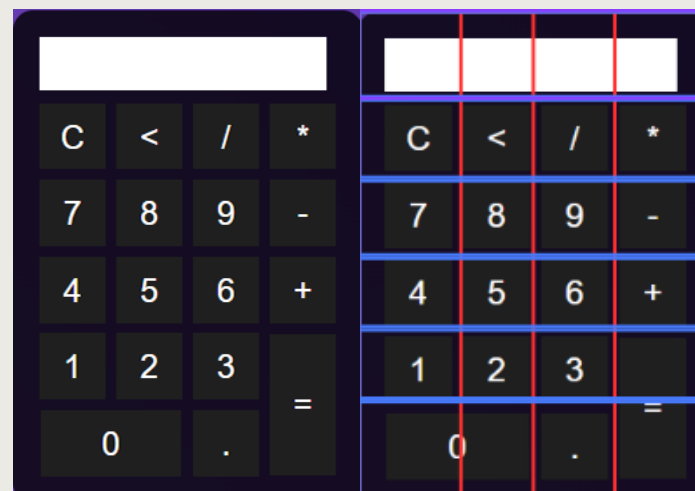
Na quarta linha, explique que é preciso usar o recurso “rowspan=2” para o botão de “=”, pois ele irá ocupar mais espaço que os outros botões.

Além da classe “botao”, é preciso adicionar uma nova classe, chamaremos de “botaoigual”

Mostre o resultado aos alunos.

Código HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Calculadora</title>
</head>

<body>
    <div class="fundo">

        <h1>Calculadora</h1>
        <div class="calculadora">

            <p id="resultado"></p>

            <table>
                <tr>
                    <td><button class="botao">C</td>
                    <td><button class="botao"><</td>
                    <td><button class="botao">/</button></td>
                    <td><button class="botao">*</button></td>
                </tr>

                <tr>
                    <td><button class="botao">7</button></td>
                    <td><button class="botao">8</button></td>
                    <td><button class="botao">9</button></td>
                    <td><button class="botao">-</button></td>
                </tr>
```

```
        <tr>
            <td><button class="botao">4</button></td>
            <td><button class="botao">5</button></td>
            <td><button class="botao">6</button></td>
            <td><button class="botao">+</button></td>
        </tr>

        <tr>
            <td><button class="botao">1</button></td>
            <td><button class="botao">2</button></td>
            <td><button class="botao">3</button></td>
            <td rowspan="2"><button class="botao botaoigual">=</button></td>

        </tr>

      </table>
    </div>
  </div>
</body>
</html>
```

Calculadora

Resultado →

C	<	/	*
7	8	9	-
4	5	6	+
1	2	3	=

Peça para que os alunos façam a última linha da tabela, alertando que eles deverão usar um pocedimento parecido com o botão de igualdade "=" para o número "0", mas dessa vez utilizando o "colspan".

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Calculadora</title>
  <link rel="stylesheet" href="style.css">
  <script src="calculadora.js"></script>
</head>
<body>
  <div class="fundo">
    <h1>Calculadora</h1>

    <div class="calculadora">
      <p id="resultado">0</p>

  <table>
        <tr>
          <td><button class="botao" >C</button></td>
          <td><button class="botao" ><</button></td>
          <td><button class="botao" >/</button></td>
          <td><button class="botao" >*</button></td>
        </tr>
        <tr>
          <td><button class="botao" >7</button></td>
          <td><button class="botao" >8</button></td>
```

AGIR

```
        <td><button class="botao" >9</button></td>
        <td><button class="botao" >-</button></td>
      </tr>
      <tr>
        <td><button class="botao" >4</button></td>
        <td><button class="botao" >5</button></td>
        <td><button class="botao" >6</button></td>
        <td><button class="botao" >+</button></td>
      </tr>
      <tr>
        <td><button class="botao" >1</button></td>
        <td><button class="botao" >2</button></td>
        <td><button class="botao" >3</button></td>
        <td rowspan="2"><button class="botao botaoigual">=</button></td>
      </tr>
      <tr>
        <td colspan="2"><button class="botao" >0</button></td>
        <td><button class="botao" >.</button></td>
      </tr>
    </table>
  </div>
 </div>
</body>
</html>
```

Calculadora

0

Resultado

C	<	/	*
7	8	9	-
4	5	6	+
1	2	3	=
0		.	

AULA 10

Calculadora (Parte 2)

Materiais necessários:

Computador, projetor e internet.

Duração:

2 horas

Objetivo:

Apresentar tipos de variáveis, operadores matemáticos e condicionais

ENGAJAR

Inicie a aula apresentando a seguinte situação para os alunos:

> *"Imagine que você está desenvolvendo uma loja virtual e precisa criar um sistema que calcule automaticamente o preço final de um produto com base em descontos personalizados."*

Pergunte aos alunos:

- O que é necessário para guardar o preço inicial do produto?
- Como podemos aplicar o desconto?
- Como o sistema pode decidir se um desconto pode ou não ser aplicado?

Explique aos alunos que esses problemas poderiam ser resolvidos utilizando **Condicionais, operadores e variáveis**, assuntos que foram vistos anteriormente no scratch.

Por isso, recapitule com os alunos os conceitos anteriormente vistos.

Variáveis

Um espaço na memória do computador onde armazenamos dados que podem ser usados ou alterados. **No JS podemos identificá-las com 3 palavras reservadas:**

Var
Let
Const.

Operadores

Operadores são como ferramentas que usamos para realizar tarefas específicas no código. Eles ajudam o programa a fazer cálculos, tomar decisões, comparar coisas ou até combinar informações. Os operadores podem ser aritméticos, lógicos, de atribuição ou de comparação.

Tipo de Operador	Operador	Descrição	Exemplo	Resultado
Aritméticos	+	Soma dois valores	5 + 3	8
Aritméticos	-	Subtrai um valor de outro	10 - 7	3
Aritméticos	*	Multiplica dois valores	4 * 2	8
Aritméticos	/	Divide um valor pelo outro	8 / 2	4
Comparação	===	Verifica se dois valores e seus tipos são **exatamente iguais**	5 === 5	true
Comparação	!==	Verifica se dois valores ou tipos são **diferentes**	5 !== '5'	true
Comparação	>	Verifica se um valor é **maior** que outro	10 > 5	true
Comparação	<	Verifica se um valor é **menor** que outro.	3 < 7	true
Comparação	>=	Verifica se um valor é **maior ou igual** a outro	8 >= 8	true
Comparação	<=	Verifica se um valor é **menor ou igual** a outro	4 <= 9	true
Lógicos	&& (E)	Retorna **true** se **todas** as condições forem verdadeiras	(5 > 3) && (8 > 5)	true
Lógicos	||	Retorna **true** se **uma** das condições forem verdadeiras.		
Lógicos	! (Não)	Inverte o valor lógico (true vira false e vice-versa)	!(5 > 3)	false

Atribuição	=	Atribui um valor a uma variável	let x = 10;	x = 10
Atribuição	+=	Soma e atribui o resultado à variável	let x = 5; x += 3;	x = 8
Atribuição	-=	Subtrai e atribui o resultado à variável	let x = 10; x -= 4;	x = 6

Estrutura Condicional

As estruturas condicionais são blocos de código que permitem tomar decisões baseadas em condições utilizando os operadores lógicos e de comparação. Se uma condição for verdadeira, um conjunto de instruções é executado; caso contrário, outra ação é realizada.

INVESTIGAR

Após a explicação inicial, mostre aos alunos como criar uma variável e explique a diferença entre let e const. Let deve ser usando quando o valor da variável precisa ser modificado no decorrer do código já o const é ideal para valores que não sofrem alteração.

Explique também que nomear uma variável no JavaScript, é importante seguir boas práticas e as regras da linguagem para garantir que o código seja claro, legível e funcional, como por exemplo a variável **deve começar com uma letra, $, ou _ e não pode começar com números e ter espaços.**

Peça agora para que os alunos criem um arquivo JS na pasta da calculadora criada na aula passada. Junto aos alunos indique quais variáveis teríamos dentro de uma calculadora.

```
let valorAtual = ' ';
```

Essa variável é essencial para a interação com a interface e para exibir o valor atual na tela da calculadora sendo responsável por armazenar o número ou o valor que está sendo digitado pelo usuário. Inicialmente, ela é definida como uma string vazia, porque no início da calculadora, não há valor a ser exibido. Ela será atualizada conforme o usuário digitar números ou operadores.

```
let operador = null;
```

A variável "operador" serve para controlar a operação que deve ser realizada quando os números forem calculados. Ela armazena o operador aritmético selecionado pelo usuário (+, -, * ou /) através do clique. Ela começa com null, pois no início da operação não há operador definido.

```
let valorAnterior = null;
```

A variável valorAnterior guarda o valor antes da operação ser realizada. Ou seja, é o número que o usuário já havia digitado antes de escolher um operador. Também começa com null porque, no início, não há valor anterior a ser armazenado.

Após a definição das variáveis explique que a calculadora deverá conter funções, sendo as principais:

AdicionarValor(valor):

Onde se o campo de resultado for '0', o valor atual será substituído pelo número pressionado (valorAtual = valor), caso contrário, o número pressionado é concatenado ao valor atual (valorAtual += valor).

apagarNumero():

Tem a função de de apagar o último número digitado.

limpar():

A função limpar é chamada quando o usuário quer resetar a calculadora basicamente limpando todas as variáveis e colocando o valor da tela de volta para "0".

definirOperador(op):

Quando o usuário escolhe um operador, essa função é chamada.

calcular():

Função calcular realiza a operação aritmética, dependendo da operação escolhida.

Após a explicação da função explique aos alunos como criar a função "adicionarValor". Como visto na explicação anterior, temos que, se o valor for exatamente "0" a função deve substituir o valor pelo valor pressionado, caso um número diferente de zero esteja aparecendo, o novo valor deve ser concatenado, ou seja ele será adicionado à direita do número anterior. Por isso devemos usar o If e o Else. A função ficará dessa forma.

```
function adicionarValor(valor) {
    const resultado = document.getElementById('resultado');
    if (resultado.innerText === '0') {
        valorAtual = valor;
    } else {
        valorAtual += valor;
    }
    resultado.innerText = valorAtual;
}
```

Avançado para a função "apagarNumero", peça para que os alunos pesquisem qual propriedade no JS deve ser usada para que seja removido apenas o último número digitado.

Após um tempo de pesquisa, explique que essa ação pode ser feita utilizando o método "slice" que pode ser aplicada na variável valorAtual.

Sendo assim:

```
function apagarNumero( ) {
    const resultado = document.getElementById('resultado');
    valorAtual = valorAtual.slice(0, -1);
    resultado.innerText = valorAtual; // Mostra "0" se o valor atual for vazio
}
```

Peça agora para que os alunos desenvolvam as funções limpar() e definirOperador(op) e chamem as funções criadas no arquivo html.

Resultado final:

```
HTML
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Calculadora</title>
    <link rel="stylesheet" href="style.css">
    <script src="calculadora.js" defer></script>
</head>
<body>
    <div class="fundo">
        <h1>Calculadora</h1>

        <div class="calculadora">
            <p id="resultado">0</p>

            <table>
                <tr>
                    <td><button class="botao" onclick="limpar()">C</button></td>
                    <td><button class="botao" onclick="apagarNumero()"><</button></td>
                    <td><button class="botao" onclick="definirOperador('/')">/</button></td>
                    <td><button class="botao" onclick="definirOperador('*')">*</button></td>
                </tr>
                <tr>
                    <td><button class="botao" onclick="adicionarValor('7')">7</button></td>
                    <td><button class="botao" onclick="adicionarValor('8')">8</button></td>
                    <td><button class="botao" onclick="adicionarValor('9')">9</button></td>
                    <td><button class="botao" onclick="definirOperador('-')">-</button></td>
```

AGIR

```html
        </tr>
        <tr>
          <td><button class="botao" onclick="adicionarValor('4')">4</button></td>
          <td><button class="botao" onclick="adicionarValor('5')">5</button></td>
          <td><button class="botao" onclick="adicionarValor('6')">6</button></td>
          <td><button class="botao" onclick="definirOperador('+')">+</button></td>
        </tr>
        <tr>
          <td><button class="botao" onclick="adicionarValor('1')">1</button></td>
          <td><button class="botao" onclick="adicionarValor('2')">2</button></td>
          <td><button class="botao" onclick="adicionarValor('3')">3</button></td>
          <td rowspan="2"><button class="botao botaoigual" onclick="calcular()">=
          </button></td>
        </tr>
        <tr>
          <td colspan="2"><button class="botao botaozero" onclick="adicionarValor('0')"
          >0</button></td>
          <td><button class="botao" onclick="adicionarValor('.')">.</button></td>
        </tr>
      </table>
    </div>
  </div>
</body>
</html>
```

JS

```javascript
let valorAtual = ' ';
let operador = null;
let valorAnterior = null;

function adicionarValor(valor) {
  const resultado = document.getElementById('resultado');
  if (resultado.innerText === '0') {
    valorAtual = valor;
  } else {
    valorAtual += valor;
  }
  resultado.innerText = valorAtual;
}

function apagarNumero() {
  const resultado = document.getElementById('resultado');
  valorAtual = valorAtual.slice(0, -1);
  resultado.innerText = valorAtual; // Mostra "0" se o valor atual for vazio
}

//AGIR
function limpar() {
  valorAtual = '';
  operador = null;
  valorAnterior = null;
  document.getElementById('resultado').innerText = '0';
}

function definirOperador(op) {
  if (valorAtual === ' ') return;
  operador = op;
  valorAnterior = parseFloat(valorAtual);
  valorAtual = ' ';
  document.getElementById('resultado').innerText = ' ';
}
```

AULA 11

Calculadora (Parte 3)

Tema:

Condicionais

Materiais necessários:

Computador, projetor e internet.

Duração:

2 horas

Objetivo:

Apresentar a condicional switch no Javascript.

ENGAJAR

Nesta aula, usaremos o comando “switch” em Javascript. Pergunte aos alunos se já ouviram falar sobre este comando, por fim explique que “switch” é uma estrutura condicional que oferece várias opções de comparação, o que facilita a leitura e a manutenção do código.

Durante esta atividade, vamos trabalhar com a palavra-chave “case”. O “case” é utilizado dentro da estrutura de controle “switch” para representar possíveis valores que a expressão do “switch” pode assumir. Quando a expressão do “switch” corresponde a um dos valores definidos pelos “cases”, o bloco de código associado a esse “case” é executado. Essa estrutura permite organizar diferentes caminhos de execução com base em um valor específico, facilitando a execução de diferentes blocos de código dependendo do resultado da comparação.

```
function mudarCor() {
  const corSelecionado = document.getElementById(“Cor”).value;
  let corFundo;

  switch (corSelecionado) {
    case “vermelho”:
      corFundo = “red”;
      break;
    case “verde”:
      corFundo = “green”;
      break;
    case “azul”:
      corFundo = “blue”;
      break;
    case “rosa”:
      corFundo = “pink”;
      break;
    case “laranja”:
      corFundo = “orange”;
      break;
    case “amarelo”:
      corFundo = “yellow”;
      break;
    case “padrao”:
    default:
      corFundo = “white”; // Cor padrão
      break;
  }
```

INVESTIGAR

Após a explicação inicial, peça para que os alunos pesquisem o método toString(), esse método será utilizado na função calcular().

O método toString() é utilizado para obter uma versão em texto de um valor, no caso da nossa atividade irá converter de um valor numérico para uma string.

Exemplo:

```
let num = 123;

console.log(num.toString());
```

AGIR

Peça agora para que os alunos desenvolvam a função calcular(). Lembre--os de utilizar os conceitos aprendidos na aula de hoje!

Resultado:

```
let valorAtual = '';
let operador = null;
let valorAnterior = null;

function adicionarValor(valor) {
   const resultado = document.getElementById('resultado');
   if (resultado.innerText === '0') {
      valorAtual = valor;
   } else {
      valorAtual += valor;
   }
   resultado.innerText = valorAtual;
}

function apagarNumero() {
   const resultado = document.getElementById('resultado');
   valorAtual= valorAtual.slice(0, -1);
   resultado.innerText = valorAtual; // Mostra "0" se o valor atual for vazio
}

//AGIR
function limpar() {
   valorAtual = '';
   operador = null;
   valorAnterior = null;
   document.getElementById('resultado').innerText = '0';
}

function definirOperador(op) {
    if (valorAtual === '') return;
   operador = op;
   valorAnterior = parseFloat(valorAtual);
   valorAtual = '';
   document.getElementById('resultado').innerText = ' ';
}

function calcular() {
   if (operador === null || valorAtual === ' ') return;
   const valorAtualNum = parseFloat(valorAtual);
   let resultado;
   switch (operador) {
      case '+':
         resultado = valorAnterior + valorAtualNum;
         break;
      case '-':
         resultado = valorAnterior + valorAtualNum;
         break;
      case '*':
         resultado = valorAnterior + valorAtualNum;
         break;
      case '/':
         resultado = valorAnterior + valorAtualNum;
         break;   }
   valorAtual = resultado.toString();
   operador = null;
   valorAnterior = null;
   document.getElementById('resultado').innerText = valorAtual;
}
```

AULA 12

Calculadora (Parte 4)

Tema:

Condicionais

Materiais necessários:

Computador, projetor e internet.

Duração:

2 horas

Objetivo:

Aprofundar o conteúdo de CSS ao aplicar no desenvolvimento de uma calculadora

ENGAJAR

Nessa aula o nosso foco será na estilização da nossa calculadora, deixando-a não só funcional, mas também visualmente atraente! Para isso vamos usar algumas propriedades do CSS.

Vamos começar explorando algumas delas!

transform:
Essa propriedade nos permite transformar elementos de diversas maneiras – podemos girar, escalar, mover ou até mesmo inclinar a nossa calculadora ou seus botões.

border-radius:
Quer deixar os cantos da sua calculadora e dos botões arredondados? O border-radius vai fazer isso! Com essa propriedade, podemos suavizar os cantos de elementos e criar um visual mais moderno e elegante.

background-image: linear-gradient:
Com essa propriedade podemos criar gradientes, ou seja, transições suaves de uma cor para outra. Isso cria um efeito visual incrível e dá uma sensação de profundidade e movimento. Podemos, por exemplo, criar um gradiente que vai do azul para o verde, dando a sensação de que a calculadora está "iluminada".

INVESTIGAR

E se quisermos aplicar uma regra de estilo a todos os elementos de uma página ou da calculadora sem precisar especificar cada um individualmente?

Peça para que os alunos pesquisem como isso seria possível, e apliquem esse seletor universal em seus códigos.

```
*{
  margin: 0;
  padding: 0;
}
```

AGIR

Peça agora para que os alunos estilizem suas calculadoras fazendo uso dos conteúdos e propriedades vistos nas últimas aulas!

```
*{
   margin: 0;
   padding: 0;
}

h1{
   padding-top: 50px;
}

.fundo{
   background-image: linear-gradient(45deg, rgb(206, 84, 27), rgb(79, 42, 165));
   height: 100vh;
   color: white;
   font-family: Arial, Helvetica, sans-serif;
   text-align: center;
}

.calculadora{
   padding: 15px;
   background-color: rgba(0, 0, 0, 0.8);
   width: 235px;
   top: 50%;
   left: 50%;
   border-radius: 15px;
   transform: translate(-50%,-50%);
   position: absolute;
}

.botao{
   width: 50px;
```

```
   height: 50px;
   font-size: 25px;
   margin: 3px;
   background-color: rgb(31, 31, 31);
   border: none;
   color: white;
   cursor: pointer;
}


}
.botaoigual{
   height: 106px;
}

.botaozero{
   width: 106px;
}

.botao:hover{
   background-color: black;
}

#resultado{
   width: 207px;
   background-color: azure;
   height: 30px;
   margin: 5px;
   font-size: 25px;
   color: black;
   text-align: right;
   padding: 5px;
}
```

AULA 13

Jogo de adivinhação de números

Tema:
JavaScript

Materiais necessários:
Computador, projetor e internet.

Duração:
2 horas

Objetivo:
Aprofundar o conhecimento em JavaScript por meio da criação de um jogo de adivinhação de números.

ENGAJAR

Nesta aula, apresentaremos um novo comando. Para isso, é interessante saber um pouco mais sobre os conhecimentos dos alunos, perguntando a eles se já ouviram falar ou já utilizaram esses novos comandos.

Após isso, apresente os novos elementos que usaremos em HTML: teremos o “SELECT” junto com “OPTION”. O ‘select’ tem outros elementos e é apenas uma estrutura básica; é uma opção entre vários disponíveis, que contém uma ou mais opções. Ele é utilizado em formulários para facilitar a seleção de dados em categorias.

Já o ‘option’ representa cada opção no menu suspenso. Nele, contém o texto visível e um valor associado, que pode ser enviado quando finalizado.

HTML:

Select
Option

No javaScript temos as função ‘math’ , ‘random‘, ‘floor’ usadas um conjuntos . O math é uma função biblioteca, ela fornece funções úteis para operações matemáticas e junto com o random ele irá gerar um número aleatório entre 0 e 1. Já com o floor, ele irá arredondar um número `x` para baixo, retornando o maior inteiro que será menor ou igual a ‘x’.

JS:

Math
random

floor

Abaixo temos um pequeno exemplo usando cor aleatória:

```
<!DOCTYPE html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Cor Aleatória com Math.random() e Math.floor()</title>
  <style>
    body {
        display: flex;
        justify-content: center;
        align-items: center;
        height: 100vh;
        margin: 0;
        font-family: Arial, sans-serif;
        background-color: #f0f0f0;
    }
#quadrado {
        width: 200px;
        height: 200px;
        background-color: #3498db;
        display: flex;
        justify-content: center;
        align-items: center;
        color: white;
        font-size: 24px;
        border-radius: 10px;
        transition: background-color 0.3s ease;
    }

    button {
        margin-top: 20px;
```

```
        padding: 10px 20px;
        font-size: 16px;
        cursor: pointer;
      }
    </style>
  </head>
  <body>
    <div id="quadrado">Cor!</div>
    <div>
    <div class="name">- Mude a cor do quadrado</div>
    <button onclick="mudarCor()"> Clique aqui!</button>

    <script>
      function gerarCorAleatoria() {
        const r = Math.floor(Math.random() * 256);
        const g = Math.floor(Math.random() * 256);
       const b = Math.floor(Math.random() * 256);
        return `rgb(${r}, ${g}, ${b})`;
      }

      function mudarCor() {
        const quadrado = document.getElementById("quadrado");
        quadrado.style.backgroundColor = gerarCorAleatoria();
      }
    </script>
  </body>
</html>
```

No CSS usaremos:☐

CSS:

letter-spacing
box-shadow
    transform
    transition
    all
    ease
    ease-in-out
    Outline

Letter-spacing: esta propriedade é usada para controlar o espaçamento entre os caracteres de um texto. São úteis para melhorar a legibilidade, o design visual e criar efeitos estéticos em textos.

Box-shadow: propriedade usada para adicionar sombras nos elementos, proporcionando profundidade e um efeito tridimensional.

Transform: usada para aplicar transformações em elementos ele permite que você mova, redimensione, gire ou inclua uma perspectiva nos elementos. Essa propriedade pode ser utilizada para criar efeitos visuais dinâmicos.

Transition: é utilizada para definir como as mudanças de estilo de um elemento devem ocorrer ao longo do tempo. Uma propriedade de duração com função de tempo.

All: é utilizada na propriedade `transition` para indicar que todas as propriedades que mudam devem ser animadas.

Ease: facilita uma articulação entre uma função e outra com transição de cor numa função de tempo.

Ease-in-out: é uma função que facilita entrada e saída mantendo uma sequência em um anime.

Outline: ele é usado para definir uma linha de contorno ao redor de um elemento. Diferente da propriedade border, o outline não ocupa espaço no layout e não afeta o posicionamento de elementos.

INVESTIGAR

Vamos colocar em prática o que foi visto! Peça para que os alunos elaborem um jogo em que um número aleatório é selecionado, e o usuário deve adivinhar qual é, recebendo sempre um retorno se o número é maior ou menor.

Proposta de solução:

HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Jogo</title>
    <meta charset="utf-8">
    <link rel="stylesheet" type="text/css" href="main.css">
  </head>
  <body>
    <div>
        <label for="difficulty">Jogo de adivinhação de números!</
label>
       <input type="number" id="guessInput" placeholder="Digite um
número">
       <button id="guessButton">Adivinhar</button>
       <p id="feedback"></p>
       <p id="guesses"></p>
    </div>
  </body>
  <script type="text/javascript" src="script.js"></script>
  </html>
```

JavaScript

```
let randomNumber;
let guessesMade = 0;

const button = document.getElementById('guessButton');
const feedback = document.getElementById('feedback');
const guesses = document.getElementById('guesses');

randomNumber = Math.floor(Math.random() * 20) + 1;

button.addEventListener('click', () => {
const  guess  =  Number(document.getElementById('guessInput').
value);
guessesMade++;
if (!guess) {
  feedback.textContent = "Por favor, insira um número!";
  return;
}

if (guess === randomNumber) {
  feedback.textContent = "Parabéns, você acertou!";
  guesses.textContent = `Você realizou ${guessesMade} tentativas.`;
}else if (guess < randomNumber) {
    feedback.textContent = "Tente um número maior.";
} else {
    feedback.textContent = "Tente um número menor.";
}
});
```

AGIR

Agora, peça para que os alunos criem uma experiência visual mais atraente para os jogadores implementando o CSS no jogo.

Proposta de solução:

CSS

```css
body {
    font-family: sans-serif;
    background: linear-gradient(135deg, #354b63, #425b76);
    margin: 0;
    padding: 20px;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    color: #f5f5f5;
}
div {
    background-color: #4b6887;
    border-radius: 20px;
    box-shadow: 0 10px 25px rgba(0, 0, 0, 0.2);
    padding: 30px;
    text-align: center;
    width: 350px;
    border: 1px solid #c0c0c0;
    transition: all 0.3s ease-in-out;
}
div:hover {
    transform: translateY(-10px);
    box-shadow: 0 15px 35px rgba(0, 0, 0, 0.3);
}
label, p {
    font-size: 20px;
    margin-bottom: 15px;
}
label {
  color: #c0c0c0;
  font-weight: 500;
  letter-spacing: 1px;
}
```

```css
#feedback, #guesses{
   font-size: 17px;
   margin-top: 15px;
   font-weight: 500;
   letter-spacing: 0.8px;
}
input {
   width: 320px;
   margin-top: 5px;
   padding: 12px;
   margin-bottom: 20px;
   border: 2px solid #c0c0c0;
   border-radius: 10px;
   font-size: 17px;
   background-color: #3b4f66;
   color: #f5f5f5;
   transition: border-color 0.3s ease;
}
input:focus {
   border-color: #ffd700;
   outline: none;
}
button {
   width: 100%;
   padding: 14px;
   background-color: #ffd700;
   border: none;
   color: #1e2a38;
   font-size: 19px;
   font-weight: bold;
   border-radius: 10px;
   cursor: pointer;
   transition: background-color 0.3s ease, transform 0.3s ease;
}
button:hover {
   background-color: #e5c100;
   transform: translateY(-2px);
}
#feedback {
   padding: 10px;
   color: #ffd700;
}
#guesses {
   padding: 10px;
   color: #ffd700;
}
```

AULA 14

Jogo de adivinhação de números

Tema:
JavaScript

Materiais necessários:
Computador, projetor e internet.

Duração:
1 hora e 30 minutos

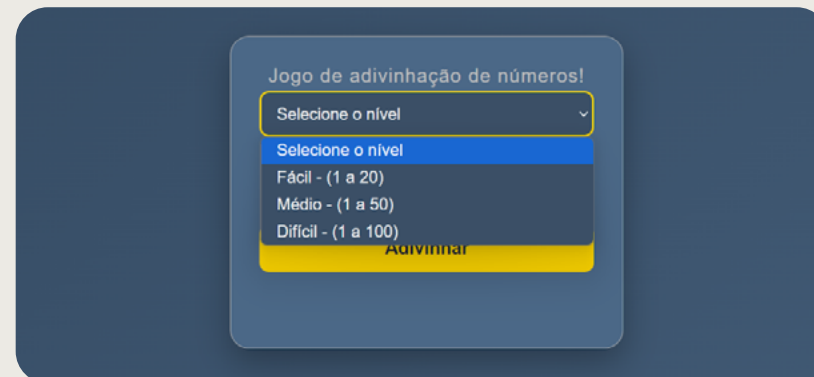
Objetivo:
Aprofundar o conhecimento em JavaScript por meio da criação de um jogo de adivinhação de números.

INVESTIGAR

Agora, toda a turma terá a oportunidade de colaborar. Juntos, os alunos deverão definir três níveis de dificuldade para o jogo (fácil, médio e difícil), especificando o intervalo de números e a quantidade de tentativas permitidas em cada nível.

Sugestão:

Fácil
Intervalo de 1 a 20
5 Tentativas
Médio
Intervalo de 1 a 50
7 Tentativas
Difícil
Intervalo de 1 a 100
10 Tentativas

AGIR

Por fim, os alunos devem aplicar as definições de níveis de dificuldade no jogo definidas anteriormente.

Proposta de solução:

Jogo de adivinhação de números!
Selecione o nível
Digite um número
Adivinhar

Jogo de adivinhação de números!
Selecione o nível
Selecione o nível
Fácil - (1 a 20)
Médio - (1 a 50)
Difícil - (1 a 100)
Adivinhar

HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Jogo</title>
    <meta charset="utf-8">
    <link rel="stylesheet" type="text/css" href="main.css">
  </head>
  <body>
    <div>
      <label for="difficulty">Jogo de adivinhação de
números!</label>
      <select id="difficulty">
        <option value="#">Selecione o nível</option>
        <option value="easy">Fácil - (1 a 20)</option>
        <option value="medium">Médio - (1 a 50)</option>
        <option value="hard">Difícil - (1 a 100)</option>
      </select>
      <input type="number" id="guessInput"
```

```
placeholder="Digite um número">
      <button id="guessButton">Adivinhar</button>
      <p id="feedback"></p>
      <p id="attempts"></p>
      <p id="guesses"></p>
    </div>
  </body>
  <script type="text/javascript" src="script.js"></script>
  </html>
```

JavaScript

```
let randomNumber;
let maxAttempts;
let attemptsRemaining;
let guessesMade;

const button = document.getElementById('guessButton');
const feedback = document.getElementById('feedback');
const attempts = document.getElementById('attempts');
const guesses = document.getElementById('guesses');

document.getElementById('difficulty').addEventListener('change',
function () {
  const difficulty = this.value;

  switch (difficulty) {
    case 'easy':
      guessesMade = 0;
      randomNumber = Math.floor(Math.random() * 20) + 1;
      maxAttempts = 5;
      break;
    case 'medium':
      guessesMade = 0;
      randomNumber = Math.floor(Math.random() * 50) + 1;
      maxAttempts = 7;
      break;
    case 'hard':
      guessesMade = 0;
      randomNumber = Math.floor(Math.random() * 100) + 1;
      maxAttempts = 10;
      break;
  }

  attemptsRemaining = maxAttempts;
  feedback.textContent = ' ';
  attempts.textContent = `Tentativas restantes:
${attemptsRemaining}`;
  guesses.textContent = ' ';
});

button.addEventListener('click', () => {
  const guess = Number(document.getElementById('guessInput').
value);

  if (!guess) {
    feedback.textContent = "Por favor, insira um número!";
    return;
```

```
  }
  guessesMade++;
  attemptsRemaining--;

  if (guess === randomNumber) {
    feedback.textContent = "Parabéns, você acertou!";
    attempts.textContent = ' ';
    guesses.textContent = `Você realizou ${guessesMade}
tentativas.`;
  } else if (attemptsRemaining === 0) {
    feedback.textContent = `Você perdeu! O número era
${randomNumber}.`;
    attempts.textContent = ' ';
    guesses.textContent = `Você realizou ${guessesMade}
tentativas.`;
  } else if (guess < randomNumber) {
    feedback.textContent = "Tente um número maior.";
    attempts.textContent = `Tentativas restantes:
${attemptsRemaining}`;
  } else {
    feedback.textContent = "Tente um número menor.";
    attempts.textContent = `Tentativas restantes:
${attemptsRemaining}`;
  }
});
```

CSS

```
body {
  font-family: sans-serif;
  background: linear-gradient(135deg, #354b63, #425b76);
  margin: 0;
  padding: 20;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  color: #f5f5f5;
}
div {
  background-color: #4b6887;
  border-radius: 20px;
  box-shadow: 0 10px 25px rgba(0, 0, 0, 0.2);
  padding: 30px;
  text-align: center;
  width: 350px;
  border: 1px solid #c0c0c0;
  transition: all 0.3s ease-in-out;
}
div:hover {
  transform: translateY(-10px);
  box-shadow: 0 15px 35px rgba(0, 0, 0, 0.3);
}
label, p {
  font-size: 20px;
  margin-bottom: 15px;
}
label {
  color: #c0c0c0;
```

```
    font-weight: 500;
    letter-spacing: 1px;
}
#feedback, #attempts, #guesses {
    font-size: 17px;
    margin-top: 15px;
    font-weight: 500;
    letter-spacing: 0.8px;
}
input{
    width: 320px;
}
select{
    width: 350px;
}
select, input {
    margin-top: 5px;
    padding: 12px;
    margin-bottom: 20px;
    border: 2px solid #c0c0c0;
    border-radius: 10px;
    font-size: 17px;
    background-color: #3b4f66;
    color: #f5f5f5;
    transition: border-color 0.3s ease;
}
select:focus, input:focus {
    border-color: #ffd700;
    outline: none;
}
button {
    width: 100%;
    padding: 14px;
    background-color: #ffd700;
    border: none;
    color: #1e2a38;
    font-size: 19px;
    font-weight: bold;
    border-radius: 10px;
    cursor: pointer;
    transition: background-color 0.3s ease, transform 0.3s ease;
}
button:hover {
    background-color: #e5c100;
    transform: translateY(-2px);
}
#feedback {
    padding: 10px;
    color: #ffd700;
}
#guesses{
    padding: 10px;
    color: #ffd700;
}
#attempts {
    color: #c0c0c0;
}
```

LÓGICA DE **PROGRAMAÇÃO** COM **C**

Ed-mundo

Realização

thp The Human Project

Apoio

SECRETARIA DE ESTADO DO DESENVOLVIMENTO ECONÔMICO E DA CIÊNCIA E TECNOLOGIA

SERGIPE GOVERNO DO ESTADO

Parceria

ATLAS RENEWABLE ENERGY

COORDENADOR Marcos Vinicius Tavares Souza

AUTORES Tabatha Costa Bastos
Juliana da Silva Firmo
Tobias da Rodrigues Silva
Isaque Rodrigues Rocha
Vitória Sales de Jesus
Kawan Lucas Silva dos Santos
Mirelly Araújo Dos Santos
Renata Santos Celestino
Josué Pereira Da Crus
Marcos Vinicius Tavares Souza

REVISÃO Mariana Carolina de Almeida Souza

PROJETO GRÁFICO Patricia Karin S. Faber

MÓDULO 1

LINGUAGEM C

AULA 01

Introdução à linguagem e tipos de variáveis.

Materiais necessários:

Notebook e Internet.

Duração:

2 horas

Objetivo:

Introduzir à linguagem C no contexto da lógica de programação. O aluno deverá sair da aula com a noção do código base em C, tais como os tipos de variáveis e a usabilidade de todos os operadores lógicos, aritméticos e de comparação.

ENGAJAR

1. Introdução à linguagem de programação e as funcionalidades do C

Vivemos na era tecnológica e a programação está por trás de tudo isso. Entender o básico dessas linguagens não é só para quem quer ser um gênio em tecnologia, mas também para todos nós, que queremos entender como as coisas funcionam no nosso dia a dia. A tecnologia avança rápido porque as pessoas estão sempre procurando soluções para os problemas que enfrentamos. Os programadores usam a lógica de programação para criar instruções que os computadores podem entender e seguir.

Num mundo cheio de diferentes linguagens de programação, a C se destaca como uma das mais importantes. Criada nos anos 70, ela é como a base para entender como a programação funciona. É fácil de aprender, mas ao mesmo tempo, oferece um monte de funcionalidades para quem está programando.

A linguagem C tem algumas características especiais. Uma delas é a "portabilidade", que significa que programas feitos em C podem funcionar em muitos tipos diferentes de computadores sem precisar mudar muita coisa. Além disso, a C é conhecida por ser boa em gerenciar a memória do computador e por ser flexível, permitindo que os programadores controlem diretamente partes importantes do sistema.

As coisas que você pode fazer com a linguagem C são incríveis! Desde criar sistemas operacionais até fazer programas para dispositivos pequenos, como os que estão dentro dos brinquedos eletrônicos. A C é como uma ferramenta que serve para muitas coisas, desde coisas bem técnicas, como fazer com que o hardware funcione, até coisas mais divertidas, como criar jogos.

Em resumo, ao aprender sobre a linguagem C, você não está apenas aprendendo a escrever códigos, está aprendendo uma linguagem que faz parte do mundo tecnológico em que vivemos. A lógica de programação e as características especiais da C nos ajudam a enfrentar os desafios do mundo digital, onde a programação não é apenas uma habilidade técnica, mas uma forma de falar a língua da tecnologia que está moldando o nosso futuro.

2. Estrutura básica da linguagem C

A linguagem C tem uma estrutura específica.

Função principal: todo seu código está dentro de uma função (funções são como pequenos blocos de códigos) principal chamada "main", é como o ponto de partida do programa, onde as coisas começam.

Declarações de variáveis: variáveis são espaços na memória onde a gente armazena alguma informação.

Mostrar na tela e receber do teclado: as duas funções responsáveis por essas coisas é a "printf" e "scanf" respectivamente.

Bibliotecas: são conjuntos de códigos pré-escritos que podem ser utilizados para agilizar o desenvolvimento de software. Elas contêm funções, classes, métodos e outros componentes que podem ser reutilizados em diferentes projetos.

```
1  #include <stdio.h>
2  int main()
3  {
4      return 0;
5  }
```

INVESTIGAR

1. Atividade de pesquisa

Podemos armazenar diversos tipos de informações diferentes. Por isso existe os tipos de variáveis da linguagem C. Portanto o aluno deverá pesquisar quais tipos de variáveis existem e relatar as principais características de cada.

2. Hello World!

No mundo dos programadores é tradição você iniciar todo e qualquer estudo de programação em alguma linguagem realizando o Hello World. Portanto, o primeiro código será o Hello World em C.

```
1  #include <stdio.h>
2  int main()
3  {
4      printf("Hello World");
5      return 0;
6  }
```

3. Tipos de variáveis

Com a pesquisa feita sobre os tipos de variáveis, serão mostradas todas as declarações e tipos de variáveis que existem na linguagem C.

char: guarda um caracter.

int: guarda um número inteiro.

float: guarda um número real com certa precisão.

double: guarda um número real com precisão maior que float.

void: tipo vazio.

Existem algumas variações destes tipos, que são:

unsigned char: caracter sem sinal.

long int: número inteiro, com domínio estendido.

unsigned int: número inteiro positivo.

unsigned long int: número inteiro positivo com domínio estendido.

short int: inteiro com domínio reduzido.

unsigned short int: inteiro positivo com domínio reduzido.

4. Ler e escrever variáveis

A função que é usada para printar alguma informação na tela é "print" e a função de receber as informações do teclado é "scanf".

```
1  #include <stdio.h>
2  int main()
3  {
4      int a;
5      scanf("%d",&a);
6      printf("O número digitado foi: %d", a);
7      return 0;
8  }
9
```

AGIR

1. Ler e escrever variáveis: desafio

O desafio consiste em o aluno declarar 5 variáveis: D = dia, M = mês, A = ano C1 = primeiro caractere e C2 = segundo caractere. Depois você precisa informar para o usuário escrever seu dia, mês e ano de nascimento e em qual unidade federativa ele nasceu. E printar todas as informações.

Entrada

A entrada contém 3 variáveis inteiras e 2 caracteres.

Saída

Imprima a mensagem “Olá, eu nasci em ‘D’ do ‘M’ de ‘A’ no estado de/do ‘C1C2’.”

Exemplos de entrada	Exemplos de entrada
16 03 1999 C E	Olá, eu nasci em 16 do 03 de 1999 no estado do CE.
23 05 2008 S E	Olá, eu nasci em 23 do 05 de 2008 no estado de SE.

AULA 02

Estruturas condicionais e operadores de comparação, lógicos e aritméticos.

Materiais necessários:
Notebook e internet.

Duração:
2 horas

Objetivo:
O aluno deverá sair com o conhecimento da estrutura condicional IF.

ENGAJAR

Quando escrevemos programas em linguagem C, muitas vezes precisamos tomar decisões com base em certas condições. As estruturas condicionais são ferramentas poderosas que nos permitem controlar o fluxo do programa com base em se uma condição seja verdadeira ou falsa. Vamos dar uma olhada em duas estruturas condicionais fundamentais em C: “if” e “else”.

1. Estrutura IF

A estrutura “if” permite que o programa execute um bloco de código somente se uma condição especificada for verdadeira.

```
#include <stdio.h>
int main() {
    int idade;
    printf("Digite a sua idade: ");
    scanf("%d", &idade);
    if (idade >= 18) {
        printf("Você é maior de idade.\n");
    }
    return 0;
}
```

Neste exemplo, o programa solicita ao usuário que digite sua idade. Se a idade digitada for maior ou igual a 18, o programa imprimirá “Você é maior de idade.” na tela.

INVESTIGAR

Na matemática existem os sinais para realização das operações. Na programação também existem seus operadores específicos que estão divididos em operadores lógicos, aritméticos e de comparação. Com isso, o aluno deverá pesquisar quais são esses operadores e procurar suas funcionalidades.

1. Operadores aritméticos

Os operadores aritméticos são utilizados para realizar operações matemáticas básicas entre variáveis ou valores numéricos.

1.1. Operador de adição (+): realiza a adição de dois valores.

```
#include <stdio.h>
int main() {
    int a = 5;
    int b = 3;
    int soma = a + b;
    printf("A soma de %d e %d é %d\n", a, b, soma);
    return 0;
}
```

1.2. Operador de subtração (-): realiza a subtração do segundo valor do primeiro.

```
#include <stdio.h>
int main() {
    int x = 8;
    int y = 4;
    int diferenca = x - y;
    printf("A diferença entre %d e %d é %d\n", x, y, diferenca);
    return 0;
}
```

1.3. Operador de multiplicação (*): realiza a multiplicação de dois valores.

```
#include <stdio.h>
int main() {
    int a = 6;
    int b = 7;
    int produto = a * b;
    printf("O produto de %d e %d é %d\n", a, b, produto);
    return 0;
}
```

1.4. Operador de divisão (/): realiza a divisão do primeiro valor pelo segundo.

```
1  #include <stdio.h>
2  int main() {
3      float dividendo = 10.0;
4      float divisor = 3.0;
5      float quociente = dividendo / divisor;
6      printf("O quociente de %.2f dividido por %.2f é %.2f\n",
7      dividendo, divisor, quociente);
8      return 0;
9  }
```

1.5. Operador de resto (%): retorna o resto da divisão do primeiro valor pelo segundo.

```
1  #include <stdio.h>
2  int main() {
3      int num = 17;
4      int divisor = 5;
5      int resto = num % divisor;
6      printf("O resto da divisão de %d por %d é %d\n",
7      num, divisor, resto);
8      return 0;
9  }
```

AGIR

2. Calculadora: desafio

O desafio consiste em você criar uma calculadora fazendo com que o usuário digite os valores e as operações calculadas serão mostradas na tela. A calculadora deverá ter operação de adição, subtração, multiplicação, divisão, resto, potenciação (o primeiro valor será a base o segundo o expoente) e raiz quadrada (do primeiro valor que foi digitado).

Entrada

A entrada deverá conter o "V1" (primeiro valor), em seguida o usuário deverá digitar o "V2" (segundo valor) e por fim o sistema mostrará todas as operações.

Saída

A saída deverá conter o nome da operação e o valor decimal (2 casas decimais) em seguida: "OPERAÇÃO = 00.00".

Exemplos de entrada	Exemplos de entrada
2 4	SOMA = 6.00 SUBTRAÇÃO = -2.00 MULTIPLICAÇÃO = 8.00 DIVISÃO = 0.50 RESTO = 2.00 POTENCIAÇÃO = 16.00 RAIZ QUADRADA = 1,41
6 5	SOMA = 11.00 SUBTRAÇÃO = 1.00 MULTIPLICAÇÃO = 30.00 DIVISÃO = 1.20 RESTO = 1.00 POTENCIAÇÃO = 7776.00 RAIZ QUADRADA = 2.45

AULA 03

Estruturas condicionais e operadores de comparação, lógicos e aritméticos.

Materiais necessários:
Notebook e internet.

Duração:
2 horas

Objetivo:
O aluno deverá sair com o conhecimento da estrutura condicional ELSE.

ENGAJAR

1. Estrutura ELSE

A estrutura "if-else" estende a capacidade de tomar decisões, permitindo que o programa execute diferentes blocos de códigos dependendo se a condição seja verdadeira ou falsa.

```c
#include <stdio.h>
int main() {
    int idade;
    printf("Digite a sua idade: ");
    scanf("%d", &idade);
    if (idade >= 18) {
        printf("Você é maior de idade.\n");
    }else{
        printf("Você é menor de idade.\n");
    }
    return 0;
}
```

INVESTIGAR

Ainda existem os operadores lógicos e de comparação. Com isso, o aluno deverá pesquisar quais são esses operadores e procurar suas funcionalidades.

1. Operadores de comparação

Esses operadores são utilizados para comparar dois valores e produzir um resultado booleano que indica se a comparação é verdadeira ou falsa.

1.1. Operador de igualdade (==): verifica se dois valores são iguais.

```c
#include <stdio.h>
int main() {
    int a = 5;
    int b = 7;
    if (a == b) {
        printf("Os valores são iguais.\n");
    } else {
        printf("Os valores são diferentes.\n");
    }
    return 0;
}
```

1.2. Operador de desigualdade (!=): verifica se dois valores são diferentes.

```c
#include <stdio.h>
int main() {
    int x = 10;
    int y = 10;
    if (x != y) {
        printf("Os valores são diferentes.\n");
    } else {
        printf("Os valores são iguais.\n");
    }
    return 0;
}
```

1.3. Operador maior que (>): verifica se o valor da esquerda é maior que o da direita.

```c
#include <stdio.h>
int main() {
    int p = 15;
    int q = 10;
    if (p > q) {
        printf("p é maior que q.\n");
    } else {
        printf("p não é maior que q.\n");
    }
    return 0;
}
```

1.4. Operador menor que (<): verifica se o valor da esquerda é menor que o da direita.

```c
#include <stdio.h>
int main() {
    int m = 8;
    int n = 12;
    if (m < n) {
        printf("m é menor que n.\n");
    } else {
        printf("m não é menor que n.\n");
    }
    return 0;
}
```

1.5. Operador maior ou igual a (>=): verifica se o valor da esquerda é maior ou igual que o da direita.

```c
#include <stdio.h>
int main() {
    int alpha = 20;
    int beta = 20;
    if (alpha >= beta) {
        printf("alpha é maior ou igual a beta.\n");
    } else {
        printf("alpha não é maior ou igual a beta.\n");
    }
    return 0;
}
```

1.6. Operador menor ou igual a (<=): verifica se o valor da esquerda é menor ou igual que o da direita.

```
1  #include <stdio.h>
2  int main() {
3      int a = 25;
4      int b = 30;
5      if (a <= b) {
6          printf("a é menor ou igual a b.\n");
7      } else {
8          printf("a não é menor ou igual a b.\n");
9      }
10     return 0;
11 }
```

2. Operadores lógicos

2.1. Operador E (&&): retorna verdadeiro se ambas as expressões forem verdadeiras.

```
1  #include <stdio.h>
2  int main() {
3      int x = 5;
4      int y = 10;
5      if (x > 0 && y > 0) {
6          printf("Ambos x e y são números positivos.\n");
7      } else {
8          printf("Pelo menos um dos números não é positivo.\n");
9      }
10     return 0;
11 }
```

2.2. Operador OU (||): retorna verdadeiro se pelo menos uma das expressões forem verdadeiras.

```
1  #include <stdio.h>
2  int main() {
3      int a = 3;
4      int b = -2;
5      if (a > 0 || b > 0) {
6          printf("Pelo menos um dos números é positivo.\n");
7      } else {
8          printf("Ambos os números são não positivos.\n");
9      }
10     return 0;
11 }
```

2.3. Operador NÃO (!): inverte o valor de verdadeiro para falso e vice-versa.

```
1  #include <stdio.h>
2  int main() {
3      int condicao = 0;
4      if (!condicao) {
5          printf("A condição é falsa.\n");
6      } else {
7          printf("A condição é verdadeira.\n");
8      }
9      return 0;
10 }
```

1. Teste de seleção

Leia 4 valores inteiros A, B, C e D. A seguir, se B for maior do que C e se D for maior do que A, e a soma de C com D for maior que a soma de A e B e se C e D, ambos, forem positivos e se a variável A for para escreva a mensagem "Valores aceitos"; caso contrário, escreva "Valores não aceitos".

Entrada

Quatro números inteiros A, B, C e D.

Saída

Mostre a respectiva mensagem após a validação dos valores.

Exemplos de entrada	Exemplos de entrada
5 6 7 8	Valores não aceitos
2 3 2 6	Valores aceitos

AGIR

AULA 04

Controle de fluxo da aplicação.

Materiais necessários:

Notebook e Internet.

Duração:

2 horas

Objetivo:

O aluno deverá sair com os conhecimentos de controle de fluxo e todas as suas estruturas e aplicações. Conseguindo distinguir cada estrutura e de acordo com o problema decidir qual usar.

ENGAJAR

Controle de fluxo refere-se à capacidade do programador de direcionar o caminho que um programa segue durante sua execução. Essa direção é influenciada por decisões lógicas e repetições, permitindo que o código execute diferentes blocos de instruções ou que repita determinadas ações conforme necessário. Os principais elementos de controle de fluxo incluem as estruturas condicionais vistas na aula passada “if” e “else” adicionando o “swith” e estruturas de loop “for”, “while” e o “do-while”.

1. Estruturas condicionais (IF e ELSE): o controle de fluxo começa com a capacidade de tomar decisões.

2. Tomada de decisão múltipla (SWITCH): às vezes é preciso escolher entre várias opções, por isso foi criado o “switch”.

```
1  #include <stdio.h>
2  int main() {
3      char escolha;
4      printf("Escolha A, B ou C: ");
5      scanf(" %c", &escolha);
6      switch (escolha) {
7          case 'A':
8              printf("Você escolheu A.\n");
9              break;
10         case 'B':
11             printf("Você escolheu B.\n");
12             break;
13         case 'C':
14             printf("Você escolheu C.\n");
15             break;
16         default:
17             printf("Escolha inválida.\n");
18     }
19     return 0;
20 }
```

3. Estruturas de repetição

3.1. “for”: essa estrutura é usada quando se sabe quantas vezes deseja repetir a ação que estará dentro do escopo.

```
1  #include <stdio.h>
2  int main() {
3      // Imprime os números de 1 a 5
4      for (int i = 1; i <= 5; i++) {
5          printf("%d ", i);
6      }
7      return 0;
8  }
```

3.2. “while”: essa estrutura executa o bloco de códigos enquanto a afirmação considerada for verdadeira.

```
1  #include <stdio.h>
2  int main() {
3      // Imprime os números de 1 a 5
4      int i = 1;
5      while (i <= 5) {
6          printf("%d ", i);
7          i++;
8      }
9      return 0;
10 }
```

INVESTIGAR

Existem três estruturas de repetições na linguagem C, duas mostradas "for" e "while" e uma ainda não vista. O aluno deverá pesquisar qual a terceira estrutura e quais as suas principais diferenças e aplicações.

1. "do-while":

Essa estrutura parece com a "while", porém ela garante que o bloco de código seja executado pelo menos uma vez, mesmo se a condição for inicialmente falsa.

```
1  #include <stdio.h>
2  int main() {
3      // Imprime os números de 1 a 5
4      int j = 1;
5      do {
6          printf("%d ", j);
7          j++;
8      } while (j <= 5);
9      return 0;
10 }
```

2. Exemplo - calculadora:

Com os códigos prontos do último desafio, vamos refazer a calculadora usando o switch para otimizar a execução.

```
1  #include <stdio.h>
2  #include <math.h>
3  int main() {
4      double V1,V2,soma,sub,mult,divi;
5      double res,pot,raiz;
6      int sinal;
7      scanf("%lf",&V1);
8      scanf("%lf",&V2);
9      printf("Escolha a operação: 1 - SOMA\n 2 - SUBTRAÇÃO\n 3 - MULTIPLICAÇÃO\n");
10     printf("4 - DIVISÃO\n 5 - RESTO\n 6 - POTENCIAÇÃO\n 7 - RAIZ QUADRADA\n");
11     scanf("%d",&sinal);
12     soma = V1 + V2;
13     sub = V1 - V2;
14     mult = V1 * V2;
15     divi = V1 / V2;
16     res = (int)V1 % (int)V2;
17     pot = pow(V1,V2);
18     raiz = sqrt(V1);
19     switch(sinal){
20         case 1:
21             printf("SOMA = %.2lf\n", soma);
22             break;
23         case 2:
24             printf("SUBTRAÇÃO = %.2lf\n", sub);
25             break;
26         case 3:
27             printf("MULTIPLICAÇÃO = %.2lf\n", mult);
28             break;
29         case 4:
30             printf("DIVISÃO = %.2lf\n", divi);
31             break;
32         case 5:
33             printf("RESTO = %.2lf\n", res);
34             break;
35         case 6:
36             printf("POTENCIAÇÃO = %.2lf\n", pot);
37             break;
38         case 7:
39             printf("RAIZ QUADRADA = %.2lf\n", raiz);
40             break;
41         default:
42             printf("ESCOLHA INCORRETA");
43     }
44
45 }
```

1. Lanche – desafio

Com base na tabela abaixo, escreva um programa que leia o código de um item e a quantidade deste item. A seguir, calcule e mostre o valor da conta a pagar.

Código	Especificação	Preço
1	Cachorro-quente	R$ 4,00
2	X-salada	R$ 4,50
3	X-bacon	R$ 5,00
4	Torrada simples	R$ 2,00
5	Refrigerante	R$ 1,50

AGIR

Entrada

O arquivo de entrada contém dois valores inteiros correspondentes ao código e à quantidade de um item, conforme a tabela acima.

Saída

O arquivo de saída deve conter a mensagem "Total: R$" seguido pelo valor a ser pago, com 2 casas após o ponto decimal.

Exemplos de entrada	Exemplos de saída
3 2	Total: R$ 10.00
4 3	Total: R$ 6.00
2 3	Total: R$ 13.50

AULA 05

Vetores.

Materiais necessários:

Notebook e Internet.

Duração:

2 horas

Objetivo:

O aluno deverá sair com o conhecimento de vetores. Conseguindo fazer as aplicações certas e usá-los na hora correta. Declarar e atribuir valores aos vetores é o assunto principal dessa aula.

ENGAJAR

Os vetores, também conhecidos como arrays são estruturas de dados fundamentais para armazenamento e manipulação de conjuntos de elementos.

Definição: um vetor é uma coleção de elementos do mesmo tipo, armazenados em posições consecutivas de memória e identificados por índices.

1. Declaração e atribuição de um vetor

```
1  #include <stdio.h>
2  int main() {
3      int numeros[5] = {1,2,3,4,5};
4      int primeiroNumero = numero[0];
5  }
```

2. Exemplos de utilização de vetores

```
1  #include <stdio.h>
2  int main() {
3      int numeros[5] = {10, 20, 30, 40, 50};
4      // Imprime os elementos do vetor
5      printf("Elementos do vetor: ");
6      for (int i = 0; i < 5; i++) {
7          printf("%d ", numeros[i]);
8      }
9      // Atualização de um elemento do vetor
10     numeros[2] = 35;
11     // Impressão dos elementos atualizados do vetor
12     printf("\nElementos atualizados: ");
13     for (int i = 0; i < 5; i++) {
14         printf("%d ", numeros[i]);
15     }
16     return 0;
17 }
```

INVESTIGAR

Em muitos casos na vida real, é preciso saber qual o tamanho de um vetor para conseguir resolver o problema. Portanto, o aluno deverá pesquisar como faz para descobrir o tamanho de um vetor.

1. Tamanho do vetor

```
1  #include <stdio.h>
2  int main() {
3      int numeros[10] = {1,2,3,4,5};
4      int tamanho = sizeof(numeros)/sizeof(numeros[0]);
5      printf("O tamanho do vetor é: %d",tamanho);
6  }
7
```

2. Exemplo

O programa a seguir calcula a média dos elementos de um vetor de números inteiros.

```
1  #include <stdio.h>
2  int main() {
3      int numeros[5] = {10, 20, 30, 40, 50};
4      int tamanho = sizeof(numeros) / sizeof(numeros[0]);
5      int soma = 0;
6      for (int i = 0; i < tamanho; i++) {
7          soma += numeros[i];
8      }
9      float media = (float)soma / tamanho;
10     printf("A média dos elementos é: %.2f\n", media);
11     return 0;
12 }
```

AGIR

3. Soma de vetores – desafio

Dados dois vetores A e B de mesmo tamanho (espaço de 5 índices), crie um programa que calcule a soma dos elementos desses vetores e armazene o resultado em um terceiro vetor C.

Entrada

O usuário irá digitar em sequência os valores de dois vetores.

Saída

O programa deverá mostrar na tela um terceiro vetor que será a soma do vetor A + B.

Exemplos de entrada	Exemplos de entrada
1 2 3 4 5 6 7 8 9 10	7 9 11 13 15
2 4 6 8 10 12 14 16 18 20	14 18 22 26 20

BANCO DE DADOS

Ed-mundo

Realização

thp The Human Project

Apoio

SECRETARIA DE ESTADO DO DESENVOLVIMENTO ECONÔMICO E DA CIÊNCIA E TECNOLOGIA

SERGIPE GOVERNO DO ESTADO

Parceria

ATLAS RENEWABLE ENERGY

MÓDULO 1

BANCO DE DADOS

AULA 01

Introdução à Banco de Dados

Materiais necessários:
Notebook.

Duração:
2 horas

Objetivo:
Adquirir o conhecimento inicial de Banco de Dados, conhecendo as definições dos principais conceitos.

ENGAJAR

1. Conceitos básicos de um banco de dados

Um banco de dados é um sistema organizado para coletar, armazenar e gerenciar dados de forma que seja fácil recuperar, atualizar e gerenciar arquivos. Ele é projetado para oferecer uma maneira eficiente e estruturada de armazenar informações, permitindo que os usuários realizem operações como inserção, atualização, recuperação e exclusão de dados.

1.1. Dados

Definição:

Dados são fatos brutos e distintos que descrevem algo, representando eventos, objetos ou informações.

Exemplo: em um contexto de uma biblioteca, os dados podem incluir o título de um livro, o nome do autor e o número de páginas.

1.2. Registros

Definição:

Um registro é uma coleção de dados relacionados, agrupados em uma única unidade.

Exemplo: em uma tabela de alunos, um registro pode representar as informações específicas de um aluno, incluindo nome, idade e número de matrícula.

1.3. Campos:

Definição:

Campos são as categorias individuais de dados que compõem um registro.

Exemplo: dentro de um registro de aluno, os campos podem ser nome, idade e número de matrícula.

1.4. Tabelas:

Definição:

Uma tabela é uma estrutura que organiza os dados em linhas e colunas, sendo a representação visual de um conjunto de registros.

Exemplo: uma tabela de livros em uma biblioteca pode ter colunas como “Título”, “Autor” e “Número de Páginas”, onde cada linha representa um livro específico.

2. Exemplo:

Construa com os alunos uma tabela de funcionários de uma empresa para armazenar suas principais informações.

2.1. Resolução

Tabela de funcionários:

Campos:

- Nome:
- Cargo:
- Data de admissão:
- Salário:

Registros:

Registro 1:

- Nome: João Silva
- Cargo: Desenvolvedor
- Data de admissão: 01/01/2022
- Salário: R$ 5.000,00

Registro 2:

- Nome: Maria Oliveira
- Cargo: Analista de Marketing
- Data de admissão: 15/02/2022
- Salário: R$ 4.500,00

Dessa forma, a tabela de funcionários organiza os dados em uma estrutura fácil de entender, permitindo que a empresa gerencie e recupere informações sobre seus colaboradores de maneira eficiente.

ENGAJAR

É nítida a importância e as inúmeras aplicações reais dos bancos de dados na sociedade. Portanto, o aluno deverá pesquisar os casos de uso comum para banco de dados. Eles podem pensar em exemplos das suas vidas diárias ou em cenários profissionais.
Logo após essa pesquisa, peça para os alunos pesquisarem sobre os tipos de dados que existem e quais são seus principais modelos de dados.

Resultado esperado

1. Tipos de dados em bancos de dados

1.1. Números Inteiros (INTEGER):
Armazena valores inteiros. Ex.: 10, 25, -5.

1.2. Números Decimais (FLOAT ou DOUBLE):
Representa valores decimais. **Ex.:** 10.99, -3.14, 5.5.

1.3. Texto (VARCHAR ou CHAR):
Armazena strings de caracteres. Ex.: "Maria", "123 Main Street".

1.4. Data (DATE ou TIMESTAMP):
Registra datas e horários. Ex.: "2023-01-15", "2023-01-15 14:30:00".

1.5. Booleano (BOOLEAN):
Armazena valores verdadeiros ou falsos. Ex.: true, false.

2. Comparação de modelos de dados:

2.1. Modelo hierárquico:
Organizado como uma árvore e relacionamentos entre pais e filhos. Ex.: banco de dados utilizado em sistemas de arquivos.

2.2. Modelo de rede:
Utiliza grafos para representar dados e relacionamentos mais complexos que o modelo hierárquico. **Ex.:** banco de dados de rede utilizado em sistemas de bibliotecas.

2.3. Modelo relacional:
Utiliza tabelas para representar dados e relacionamentos definidos por chaves primárias e estrangeiras. Ex.: banco de dados relacional utilizado em sistemas de gestão empresarial (ERP).

AGIR

Desafio:

Construa uma tabela de dados de uma criança na hora do seu nascimento e dê exemplos de possíveis registros. Defina também os tipos de cada dado.

Possível resposta

Variável	Valor	Tipo de Dado
Nome:	João Santos	string
Data de Nascimento:	15/01/2024	string
Sexo:	Masculino	string
Peso (kg):	4,5	double

AULAS 02 e 03

Modelagem conceitual

Materiais necessários:

Notebook.

Duração:

4 horas (2 aulas)

Objetivo:

Abordar conceitos mais profundos do BD, com ênfase na modelagem do projeto. Aprender sobre chaves primárias e estrangeiras.

ENGAJAR

1. Modelagem conceitual:

A modelagem conceitual visa representar os conceitos e as relações no mundo real que são importantes para o sistema.

1.1. Entidades:

No contexto de banco de dados, uma entidade refere-se a um objeto ou conceito do mundo real que pode ser identificado, armazenado e manipulado pelos sistemas de informações. Essas entidades geralmente representam objetos tangíveis ou conceitos abstratos que desejamos rastrear no sistema. Por exemplo, em um sistema de gerenciamento de biblioteca, as entidades podem incluir "Livro", "Cliente" e "Empréstimo". Cada entidade tem características específicas, chamadas de atributos, que descrevem propriedades relevantes dessa entidade.

1.2. Atributos:

Os atributos são as características ou propriedades que descrevem uma entidade em um banco de dados. Cada entidade terá diferentes atributos associados, e esses atributos representam informações específicas sobre a entidade. Tomando o exemplo de uma entidade "Cliente" em um sistema de banco de dados para uma loja online, os atributos podem incluir "Nome", "Endereço", "E-mail" e "Número de Telefone". Em termos mais técnicos, os atributos são os campos ou colunas de uma tabela em um banco de dados relacional e têm tipos de dados específicos, como texto, número inteiro ou data, dependendo do tipo de informação que estão destinados a armazenar.

2. Modelagem lógica:

A modelagem lógica traduz a modelagem conceitual para uma representação mais próxima do modelo de dados que será implementado.

Em um banco de dados, as entidades muitas vezes têm relações entre si, refletindo conexões e interações no mundo real. A ideia de relacionamentos entre entidades é fundamental para modelar e organizar dados de forma eficiente. Os relacionamentos ajudam a evitar redundância de informações e garantem a integridade dos dados. Existem três tipos principais de relacionamentos entre entidades:

2.1. Um para Um (1:1):

Um registro em uma entidade está associado a apenas um registro em outra entidade, e vice-versa. Ex.: relação entre um "Cliente" e seu "Perfil".

2.2. Um para Muitos (1:N):

Um registro em uma entidade pode estar associado a vários registros em outra entidade, mas um registro nesta outra entidade está associado a apenas um registro na primeira entidade. Ex.: Relação entre um "Departamento" e seus "Funcionários".

2.3. Muitos para Muitos (N:M):

Um registro em uma entidade pode estar associado a vários registros em outra entidade, e vice-versa. Ex.: Relação entre "Alunos" e "Disciplinas" em um sistema de gestão escolar.

Chaves primárias e estrangeiras:

Chave Primária (PK): é um atributo ou conjunto de atributos que identifica exclusivamente cada registro em uma entidade. Garante unicidade e integridade referencial. Ex.: O "ID" de um "Cliente" em uma tabela de clientes.

Chave Estrangeira (FK): é um atributo em uma tabela que se refere à chave primária de outra tabela. Estabelece o relacionamento entre as duas tabelas. Ex: O "ID_Departamento" em uma tabela de "Funcionários" referenciando a chave primária "ID" na tabela de "Departamentos".

INVESTIGAR

Nesse contexto, existem os Diagramas de Entidade-Relacionamento (DER), que ajudam na visualização e organização do BD. Pesquise quais as principais características do DER e quais seus símbolos comuns.

Resultado Esperado

1. Diagramas de Entidade-Relacionamento (DER):

Um DER é uma representação gráfica das entidades e dos relacionamentos entre elas.

1.1. Entidades (Retângulo):
No DER, as entidades são representadas por retângulos. O nome da entidade é colocado dentro do retângulo.

1.2. Relacionamentos (Linhas):
Os relacionamentos entre entidades são representados por linhas que conectam os retângulos das entidades envolvidas. A cardinalidade (1:1, 1:N, N:M) pode ser indicada nas extremidades da linha.

1.3. Atributos (Oval ou Linha Pontilhada):
Atributos simples podem ser representados por elipses (oval). Atributos multivalorados são representados por uma linha dupla pontilhada conectada ao retângulo da entidade.

1.4. Chaves primárias (Sublinhado):
A chave primária é indicada sublinhando o atributo correspondente.

1.5. Chaves estrangeiras (Sublinhado Pontilhado):

Chaves estrangeiras são indicadas sublinhando o atributo correspondente com uma linha pontilhada.

1.6. Atributos derivados (Asterisco):
Atributos derivados são indicados com um asterisco ao lado.

1.7. Atributos multivalorados (Duplo Oval ou Losango):
Atributos multivalorados são representados por uma linha dupla conectada ao retângulo da entidade ou por um losango.

Esses símbolos são essenciais para expressar visualmente a estrutura de um banco de dados e as relações entre suas entidades. Ao criar um DER, é crucial que os símbolos sejam utilizados corretamente para garantir uma representação precisa e compreensível do modelo de dados.

AGIR

Desafio:

Solicite aos alunos que identifiquem entidades e atributos em um contexto específico, como uma biblioteca ou uma loja online.

Peçam aos alunos para criarem uma tabela simples com chaves primárias e estrangeiras com base em um cenário específico de sua escolha.

Solicite aos alunos que criem um DER simples para um sistema de gerenciamento de escola, identificando entidades, atributos e relacionamentos.

Possível resposta

Entidades e atributos:

Livro:

Atributos: *ISBN (chave primária), Título, Autor, Ano de Publicação, Gênero, Número de Páginas, Editora, etc.*

Leitor:

Atributos: *ID do Leitor (chave primária), Nome, Endereço, E-mail, etc.*

Empréstimo:

Atributos: *ID do Empréstimo (chave primária), Data de Empréstimo, Data de Devolução Prevista, Livro(s) Emprestado(s), ID do Leitor (chave estrangeira), etc.*

Autor:

Atributos: *ID do Autor (chave primária), Nome, Nacionalidade, Data de Nascimento, etc.*

Relacionamentos:

Relacionamento entre livro e autor:

Um livro pode ter um ou mais autores; um autor pode ter escrito um ou mais livros.

Relacionamento: *"Escrito por"*

Cardinalidade: *1:N (um livro tem um ou mais autores, um autor escreve um ou mais livros.)*

Relacionamento entre livro e empréstimo:

Um livro pode estar associado a vários empréstimos; um empréstimo envolve um ou mais livros.

Relacionamento: *"Empréstimo de"*

Cardinalidade: *M:N (muitos livros podem estar em muitos empréstimos, muitos empréstimos podem envolver muitos livros)*

Relacionamento entre Leitor e Empréstimo:

Um leitor pode realizar vários empréstimos; cada empréstimo é feito por um leitor.

Relacionamento: *"Realiza"*

Cardinalidade: *1:N (um leitor faz muitos empréstimos, cada empréstimo é feito por um leitor.)*

Relacionamento entre empréstimo e livro:

Cada empréstimo envolve um ou mais livros; cada livro pode estar em vários empréstimos.

Relacionamento: *"Inclui"*

Cardinalidade: *M:N (muitos livros podem estar em muitos empréstimos, muitos empréstimos podem envolver muitos livros.)*

AULAS 04 e 05

Banco de dados relacional e não relacional

Materiais necessários:
Notebook.

Duração:
4 horas (2 aulas)

Objetivo:
Apresentar ao aluno a diferença entre os bancos de dados relacionais e não relacionais. Demonstrar as principais vantagens e desvantagens dos BD relacional e não relacional.

ENGAJAR

1. Princípios dos bancos de dados relacionais:

Os bancos de dados relacionais organizam dados em tabelas, que consistem em linhas e colunas.

1.1. Características principais:

- Cada tabela representa uma entidade ou conceito.
- Cada coluna na tabela representa um atributo específico da entidade.
- Cada linha contém uma instância ou registro único da entidade.

Ex.:

ID	Nome	Idade
1	João Silva	25
2	Maria Oliveira	30
3	Pedro Santos	28

2. Capacidade de estabelecer relacionamentos entre tabelas:

Os bancos de dados relacionais permitem estabelecer relacionamentos entre diferentes tabelas, conectando-as por meio de chaves.

2.1. Características principais:

Chave Primária (Primary Key - PK): Identifica exclusivamente cada registro em uma tabela.

Chave Estrangeira (Foreign Key - FK): Referencia a chave primária em outra tabela, estabelecendo uma relação entre elas.

Ex.:

Tabela: Clientes

ID	Nome	E-mail
1	Ana Oliveira	ana@email.com
2	Carlos Silva	carlos@email.com

Tabela: Pedidos

ID	ClienteID	Valor
101	1	150.00
102	2	200.00

3. Vantagens dos bancos de dados relacionais:

3.1. Integridade referencial:

A utilização de chaves primárias e estrangeiras garante a integridade referencial, mantendo a consistência dos dados entre tabelas relacionadas.

3.2. Estrutura tabular organizada:

A organização em tabelas facilita a compreensão e o acesso aos dados, permitindo uma modelagem clara e eficiente.

3.3. Facilidade de consulta e manipulação:

A linguagem SQL fornece uma interface poderosa e padronizada para consultas e manipulação de dados, permitindo a execução de operações complexas de forma eficiente.

3.4. Suporte a transações:

Os bancos de dados relacionais suportam transações ACID (Atomicidade, Consistência, Isolamento, Durabilidade), garantindo a confiabilidade e a consistência dos dados.

3.5. Normalização para redução de redundância:
O processo de normalização ajuda a reduzir a redundância e a complexidade dos dados, evitando inconsistências.

4. Desvantagens dos bancos de dados relacionais:

4.1. Desempenho em escala:
Em cenários de grande escala e alta concorrência, os bancos de dados relacionais podem enfrentar desafios de desempenho, especialmente em comparação com soluções NoSQL.

4.2. Esquema fixo:
O esquema rígido das tabelas pode tornar difícil a adaptação a mudanças nos requisitos, exigindo alterações no esquema.

4.3. Modelagem complexa para dados não estruturados:
Modelar dados não estruturados ou semiestruturados pode ser menos eficiente em um ambiente puramente relacional.

4.4. Custo de manutenção:
O gerenciamento de esquemas, índices e consultas complexas pode exigir mais esforço e recursos em comparação com sistemas mais simples.

4.5. Escalabilidade horizontal limitada:
A escalabilidade horizontal (adicionar mais servidores) pode ser mais desafiadora em comparação com algumas soluções NoSQL.

4.6. Complexidade para hierarquias profundas:
Modelar dados com hierarquias profundas (árvore ou grafo complexo) pode ser mais complexo e exigir consultas complexas.

INVESTIGAR

Para iniciar um projeto, todo profissional estrutura seu banco de dados para auxiliar nessa construção, esse processo se chama modelagem. O aluno deverá pesquisar como é feita uma modelagem de um banco de dados relacional e não relacional, dando exemplos deles.
Logo após essa pesquisa, o aluno deverá buscar o conceito de normalização de dados em BD relacionais e trazer suas principais características.

1. Normalização em bancos de dados relacionais:
A normalização é um processo utilizado em bancos de dados relacionais para organizar e estruturar dados de maneira eficiente, reduzindo a redundância e melhorando a integridade dos dados. Esse processo visa evitar anomalias de atualização, inserção e exclusão, garantindo a consistência dos dados ao longo do tempo. A normalização é aplicada a tabelas em um banco de dados para minimizar a duplicação de informações e facilitar a manutenção.
Existem várias formas normais, sendo as mais comuns a 1ª, 2ª e 3ª Formas Normais (1NF, 2NF, 3NF).

- **1ª Forma Normal (1NF):**
- **Objetivo:** garantir que cada célula em uma tabela contenha um único valor e que os valores sejam atômicos.
- **Exemplo:** uma tabela de alunos onde um aluno pode ter múltiplas disciplinas deve ser dividida em duas tabelas distintas.
- **2ª Forma Normal (2NF):**
- **Objetivo:** Alcançar a 1NF e garantir que todos os atributos não chave estejam totalmente dependentes da chave primária.
- **Exemplo:** Se uma tabela tem uma chave composta (duas ou mais colunas como chave primária), e algum atributo depende apenas de uma parte dessa chave, ele deve ser movido para uma tabela separada.

- **3ª Forma Normal (3NF):**
- **Objetivo:** Alcançar a 2NF e garantir que não haja dependência transitiva, ou seja, que nenhum atributo não chave depende de outro atributo não chave.
- **Exemplo:** Se um atributo não chave depende de outro atributo não chave, ambos devem ser movidos para uma tabela separada.

1.1. Ex.:

Considere a seguinte tabela inicial:

Tabela: Alunos

ID	Nome	Disciplina	Professor
1	João	Matemática	Silva
2	Maria	História	Oliveira
3	Pedro	Matemática	Silva

Para normalizá-la:

Aplicando a 1NF: cada célula contém um único valor atômico.

Tabela: Alunos

ID	Nome
1	João
2	Maria
3	Pedro

Tabela: Disciplinas

ID	Disciplina	Professor
1	Matemática	Silva
2	História	Oliveira
3	Matemática	Silva

Aplicando a 2NF: garantir que todos os atributos não chave estejam totalmente dependentes da chave primária.

Tabela: Alunos

ID	Nome
1	João
2	Maria
3	Pedro

Tabela: Disciplinas

ID	Disciplina	Professor
1	Matemática	Silva
2	História	Oliveira

Tabela: Alunos_Disciplinas

ID	AlunoID	DisciplinaID
1	1	1
2	2	2
3	3	1

Aplicando a 3NF: garantir que não haja dependência transitiva entre atributos não chave.

Tabela: Alunos

ID	Nome
1	João
2	Maria
3	Pedro

Tabela: Professores

ID	Nome
1	Silva
2	Oliveira

Tabela: Disciplinas

ID	Nome
1	Matemática
2	História

Tabela: Alunos_Disciplinas

ID	AlunoID	DisciplinaID
1	1	1
2	2	2
3	3	1

Desafio 01:
Crie uma tabela com uma chave primária e estabeleça uma relação com outra tabela usando uma chave estrangeira.
Desafio 02:
Mostre como aplicar a normalização a uma tabela, destacando as etapas para atingir pelo menos a terceira forma normal (3NF).

Desafio 01:
Crie uma tabela com uma chave primária e estabeleça uma relação com outra tabela usando uma chave estrangeira.
Desafio 02:
Mostre como aplicar a normalização a uma tabela, destacando as etapas para atingir pelo menos a terceira forma normal (3NF).

Possível resposta

Considere a tabela abaixo, que representa informações sobre livros e seus autores:

ISBN	Título	Autor	Ano de Publicação
1234567890	"Aventuras XYZ"	John Doe	2000
9876543210	"Segredos ABC"	Jane Smith	2015
1122334455	"Mistérios 123"	John Doe	2010

Normalização até a Terceira Forma Normal (3NF):

1. **Primeira Forma Normal (1NF):**
    - Cada célula da tabela deve conter apenas um valor.
    - Valores devem ser atômicos.

    A tabela fornecida já está na 1NF, pois cada célula contém apenas um valor e todos os valores são atômicos.

2. **Segunda Forma Normal (2NF):**
    - Atender aos requisitos da 1NF.
    - Remover dependências parciais.

    Neste caso, a dependência parcial não é um problema, pois a chave primária (ISBN) já cobre todo o conjunto de atributos.

AGIR

3. Terceira Forma Normal (3NF):

- Atender aos requisitos da 2NF.
- Remover dependências transitivas.

A tabela atual tem uma dependência transitiva, pois o atributo "Autor" depende da chave primária "ISBN". Para atingir a 3NF, devemos remover essa dependência transitiva, criando uma tabela separada para os autores.

Tabelas Normalizadas:

Tabela Livros:

ISBN	Título	Ano de Publicação
1234567890	"Aventuras XYZ"	2000
9876543210	"Segredos ABC"	2015
1122334455	"Mistérios 123"	2010

Tabela Autores:

AutorID	Autor
1	John Doe
2	Jane Smith

AULAS 06 e 07

Consultas com SQL

Materiais necessários:
Notebook.

Duração:
4 horas (2 aulas)

Objetivo:
Apresentar a linguagem SQL e seus principais comandos.

ENGAJAR

1. Sintaxe básica do SQL:

SQL (Structured Query Language) é uma linguagem padronizada para interagir com bancos de dados relacionais.

1.1. Comandos básicos SQL:

SELECT - Recuperar dados:

```
SELECT coluna1, coluna2, …
FROM nome_da_tabela
WHERE condição;
```

Ex.: Selecionar todos os alunos com idade maior que 20

```
SELECT Nome, Idade
FROM Alunos
WHERE Idade > 20;
```

INSERT - Inserir dados:

```
INSERT INTO nome_da_tabela (coluna1, coluna2, ...)
VALUES (valor1, valor2, ...);
```

Ex.: Inserir um novo aluno

```
INSERT INTO Alunos (Nome, Idade)
VALUES ('Joana', 22);
```

UPDATE - Atualizar dados:

```
UPDATE nome_da_tabela
SET coluna1 = valor1, coluna2 = valor2, …
WHERE condição;
```

Ex.: Atualizar a idade do aluno João para 26

```
UPDATE Alunos
SET Idade = 26
WHERE Nome = 'João';
```

DELETE - Excluir dados:

```
DELETE FROM nome_da_tabela
WHERE condição;
```

Ex.: Excluir todos os alunos com idade menor que 18

```
DELETE FROM Alunos
WHERE Idade < 18;
```

Esses comandos são fundamentais para interagir com um banco de dados relacional. O comando SELECT é usado para recuperar dados, o INSERT para adicionar novos registros, o UPDATE para modificar registros existentes e o DELETE para remover registros com base em condições específicas. Ao usar esses comandos de maneira adequada, os usuários podem manipular eficientemente os dados em seus bancos de dados.

2. Estrutura de uma consulta SQL e cláusulas:

Uma consulta SQL é uma instrução que é utilizada para recuperar, modificar ou interagir com dados em um banco de dados relacional. A estrutura básica de uma consulta SQL envolve diversas cláusulas que permitem personalizar e refinar os resultados desejados.

SELECT: especifica as colunas que devem ser incluídas no resultado da consulta.

Ex.:

```
SELECT Nome, Idade
FROM Alunos;
```

FROM: indica a tabela ou tabelas das quais os dados serão recuperados.

Ex.:

```
SELECT Nome, Idade
FROM Alunos;
```

WHERE: filtra os resultados com base em uma condição específica.

Ex.:

```
SELECT Nome, Idade
FROM Alunos
WHERE Idade > 20;
```

ORDER BY: classifica os resultados em ordem crescente ou decrescente com base em uma ou mais colunas.

Ex.:

```
SELECT Nome, Idade
FROM Alunos
ORDER BY Idade DESC;
```

GROUP BY: agrupa os resultados com base em uma ou mais colunas, geralmente utilizada em conjunto com funções de agregação (como COUNT, SUM, AVG).

Ex.:

```
SELECT Disciplina, COUNT(*) as Total_Alunos
FROM Alunos
GROUP BY Disciplina;
```

HAVING: filtra os resultados de uma cláusula GROUP BY com base em uma condição específica.

Ex.:

```
SELECT Disciplina, COUNT(*) as Total_Alunos
FROM Alunos
GROUP BY Disciplina
HAVING COUNT(*) > 5;
```

INVESTIGAR

Para manipulação de dados usamos a linguagem SQL. O aluno deverá pesquisar qual a estrutura básica de um comando SQL e como executá-lo.

Resultado esperado

1. Exercícios:

1.1. Seleção: selecione o nome e a nota de todos os alunos com idade superior a 25 anos.

Solução:

```
SELECT Nome, Nota
FROM Alunos
WHERE Idade > 25;
```

1.2. Inserção: adicione um novo aluno à tabela com o nome "Mariana", idade 30, e nota 88.

Solução:

```
INSERT INTO Alunos (Nome, Idade, Nota)
VALUES ('Mariana', 30, 88);
```

1.3. Atualização: atualize a nota do aluno com o nome "Pedro" para 95.

Solução:

```
UPDATE Alunos
SET Nota = 95
WHERE Nome = 'Pedro';
```

1.4. Exclusão: exclua todos os alunos com notas abaixo de 60.

Solução:

```
DELETE FROM Alunos
WHERE Nota < 60;
```

AGIR

Desafio 01:

Selecione o nome e a idade dos três alunos mais jovens, ordenados por idade.

Resultado esperado

Para executar a consulta mencionada, precisamos ter informações específicas sobre a estrutura da tabela de alunos e o esquema de dados. Vou fornecer uma consulta genérica usando a sintaxe SQL para ilustrar o processo. Assumirei que há uma tabela chamada Alunos com colunas como Nome, Idade e Nota.

Seleção dos três alunos mais jovens:

```
SELECT Nome, Idade
FROM Alunos
ORDER BY Idade
LIMIT 3;
```

Desafio 02:

Conte quantos alunos têm a mesma nota e apresente o resultado.

Resultado esperado

Esta consulta selecionará os nomes e idades dos três alunos mais jovens da tabela Alunos, ordenados por idade.

Contagem de alunos com a mesma nota:

```
SELECT Nota, COUNT(*) AS QuantidadeDeAlunos
FROM Alunos
GROUP BY Nota
HAVING COUNT(*) > 1;
```

Esta consulta conta quantos alunos têm a mesma nota e apresenta o resultado. A cláusula GROUP BY agrupa os alunos pela nota, e a cláusula HAVING filtra os grupos para incluir apenas aqueles com mais de um aluno.

DESENVOLVIMENTO **MOBILE** COM **FLUTTERFLOW**

Ed-mundo

Realização

The Human Project

Apoio

SECRETARIA DE ESTADO DO DESENVOLVIMENTO ECONÔMICO E DA CIÊNCIA E TECNOLOGIA

SERGIPE GOVERNO DO ESTADO

Parceria

ATLAS RENEWABLE ENERGY

COORDENADOR Marcos Vinicius Tavares Souza

AUTORES Tabatha Costa Bastos
Juliana da Silva Firmo
Tobias da Rodrigues Silva
Isaque Rodrigues Rocha
Vitória Sales de Jesus
Kawan Lucas Silva dos Santos
Mirelly Araújo Dos Santos
Renata Santos Celestino
Josué Pereira Da Crus
Marcos Vinicius Tavares Souza

REVISÃO Mariana Carolina de Almeida Souza

PROJETO GRÁFICO Patricia Karin S. Faber

MÓDULO 1

BANCO DE DADOS

AULA 01

Introdução a FlutterFlow

Materiais necessários:
Notebook, acesso à internet.

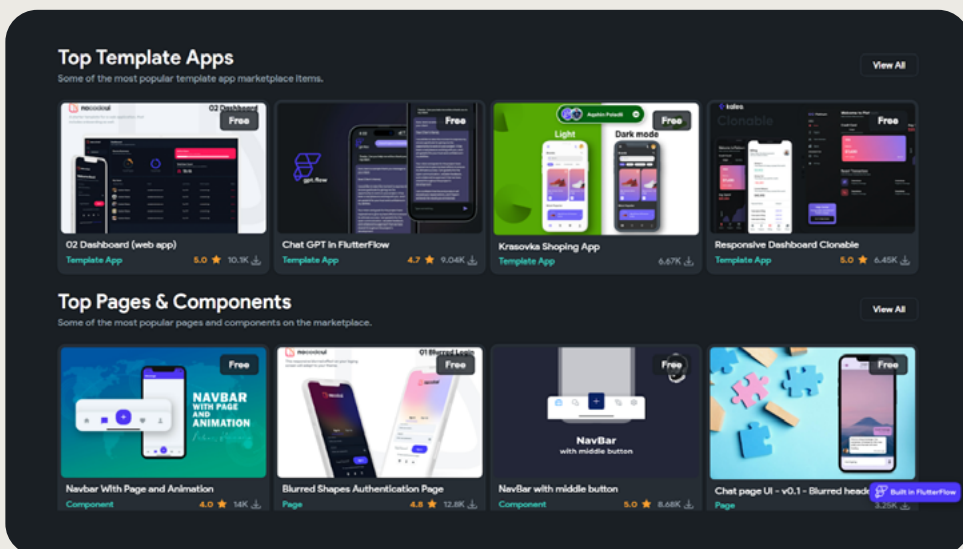
Duração:
2 horas

Objetivo:
Conhecer o FlutterFlow.

ENGAJAR

O FlutterFlow é uma ferramenta que permite criar aplicativos Flutter de forma visual e sem a necessidade de escrever código. Com o FlutterFlow, você pode projetar a interface do usuário, adicionar funcionalidades e até mesmo conectar-se a serviços externos, tudo através de uma interface intuitiva.

O que é FlutterFlow?

FlutterFlow é uma plataforma low code, ou seja, com baixo código que permite construir aplicativos de forma incrivelmente rápida no seu navegador. Com o FlutterFlow, você pode criar aplicativos totalmente funcionais com integração ao Firebase, suporte a API, animações e muito mais. É uma ferramenta poderosa para desenvolvedores que desejam criar aplicativos móveis de maneira eficiente e eficaz.

O que podemos fazer com o FlutterFlow?

Com o FlutterFlow, você pode fazer várias coisas, incluindo:

- Construir aplicativos móveis totalmente funcionais de forma rápida e eficiente.
- Integrar seu aplicativo com o Firebase para recursos como autenticação de usuários, armazenamento de dados em tempo real e notificações push.
- Adicionar animações e transições suaves aos elementos do seu aplicativo.
- Personalizar o design do seu aplicativo com uma ampla variedade de componentes e widgets disponíveis.
- Conectar seu aplicativo a APIs externas para obter dados e funcionalidades adicionais.
- Aceitar pagamentos dentro do seu aplicativo.
- Colaborar com outros membros da equipe no desenvolvimento do aplicativo.
- Exportar o código-fonte do seu aplicativo para continuar o desenvolvimento em outras ferramentas, se necessário.

Essas são apenas algumas das possibilidades que o FlutterFlow oferece. Com essa plataforma, você tem a flexibilidade e a facilidade de criar aplicativos móveis incríveis de acordo com suas necessidades e requisitos específicos.

Empresas que usam o FlutterFlow:

Alibaba	Baidu	Groupon	Hamilton	Square	Betterment	Bytedance
iRobot	Realtor.com	Kotak	CrowdSource	GooglePay	Philips Hue	CrowdSource

Essas são apenas algumas das empresas que utilizam o FlutterFlow para construir seus aplicativos móveis.

O FlutterFlow é uma ferramenta poderosa que permite criar aplicativos nativos de forma rápida e eficiente.

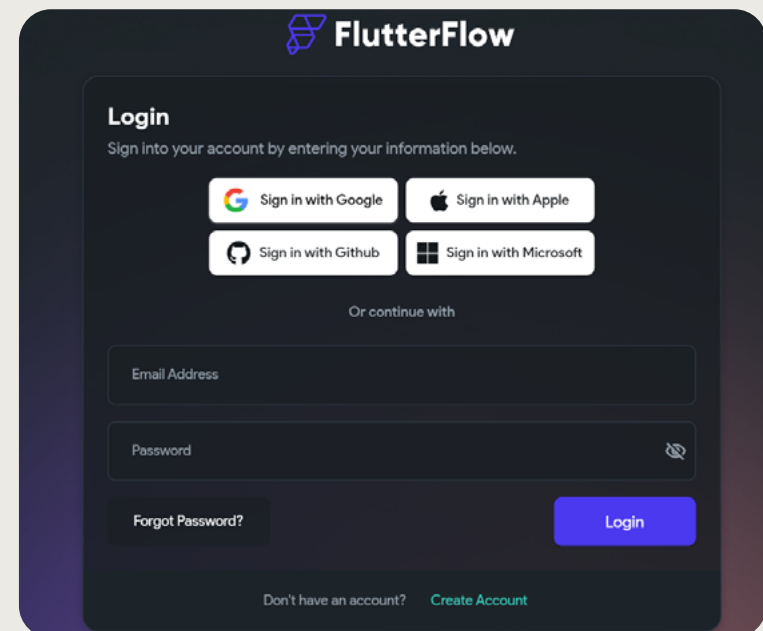
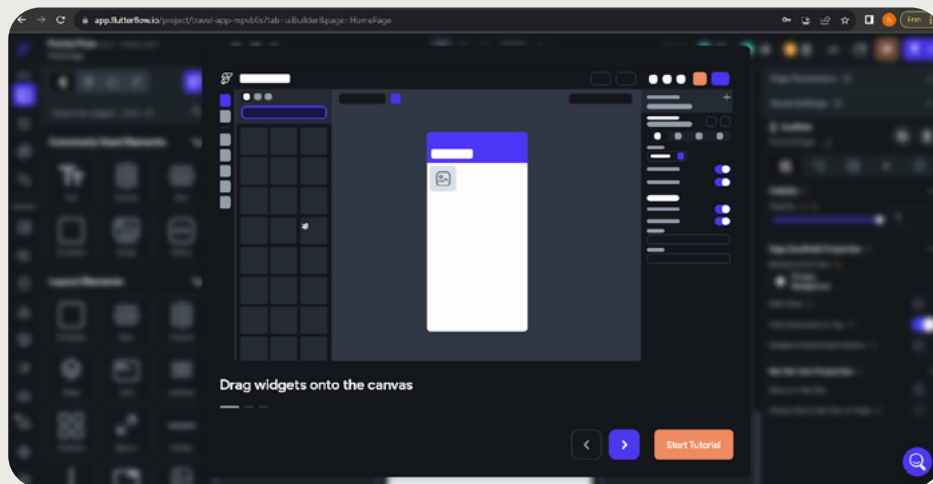
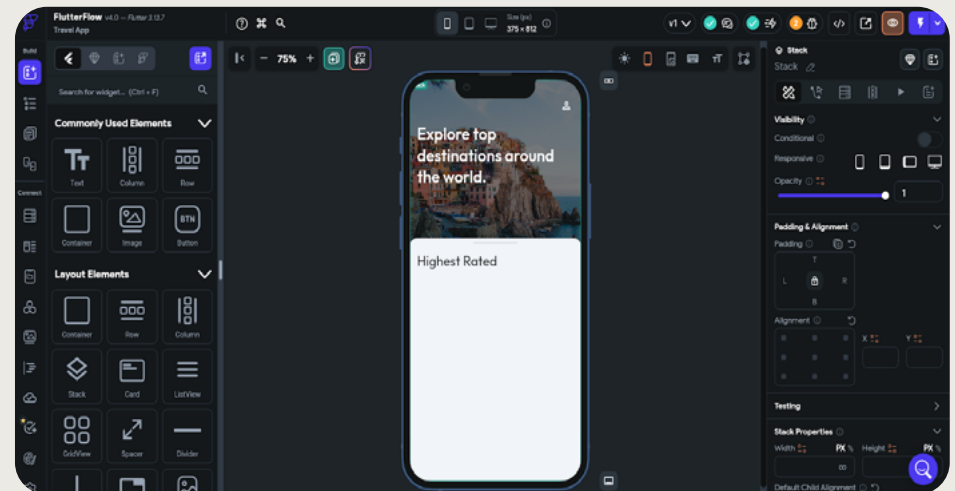
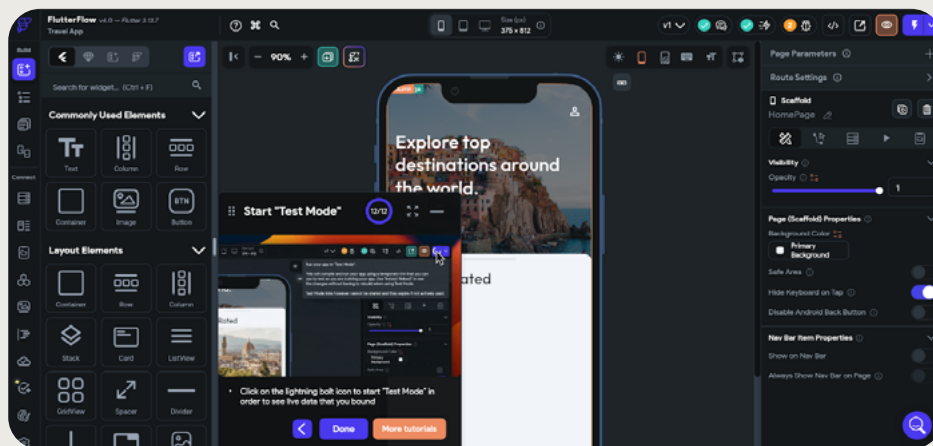
INVESTIGAR

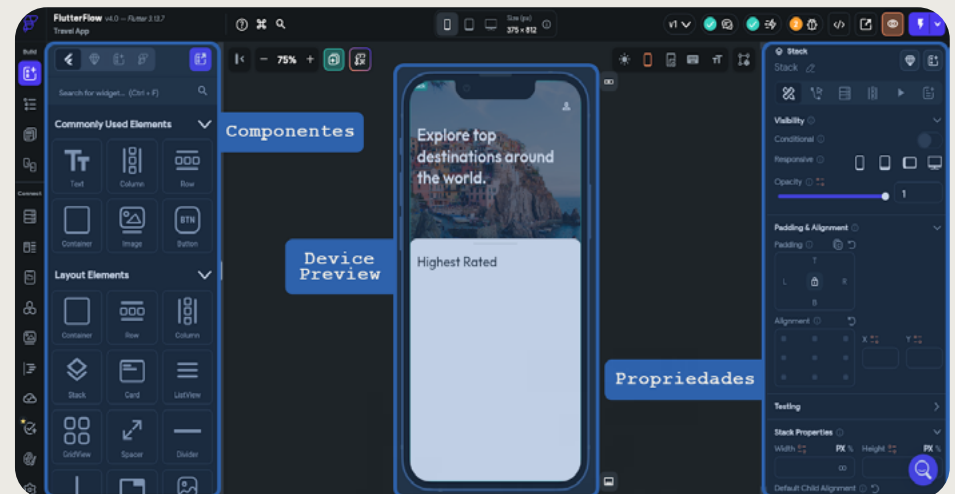
Guie os alunos a criarem uma conta no site oficial da FlutterFlow, a mesma que será utilizada durante todo o curso.

Criando uma conta:

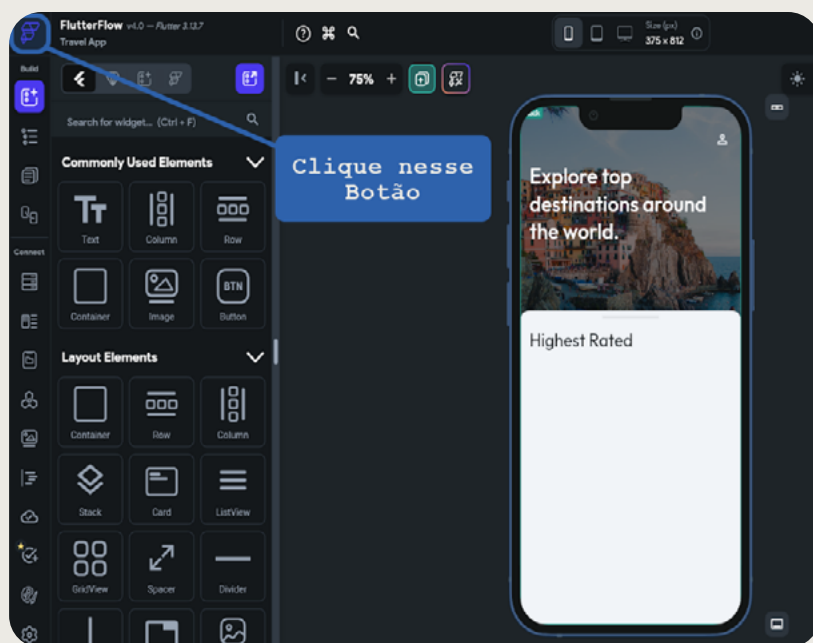
Entre no site oficial da FlutterFlow (https://flutterflow.io/).

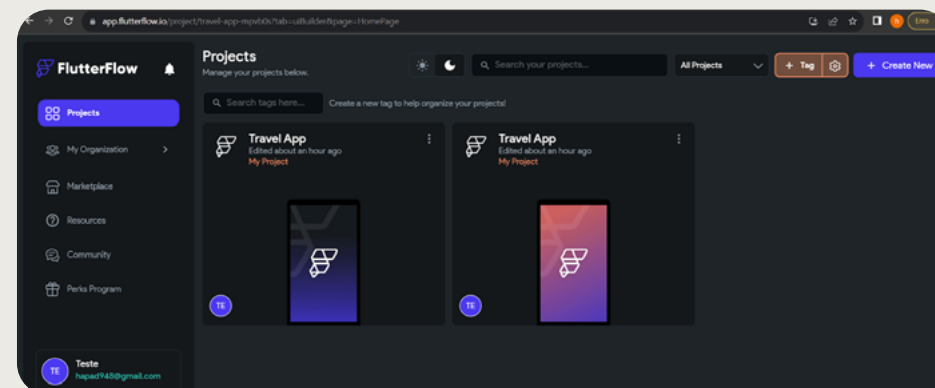
Se já possuir uma conta clique em sign. in (entrar), mas se for criar uma nova conta clique em start for free (comece de graça).

Clicando em comece de graça, você será redirecionado para a criação da conta, para criação da conta precisa somente do nome, e-mail e senha.

Ao criar a conta inicialmente aparecerá pequenas dicas, podemos clicar no botão próximo ou em **start tutorial** para começar um pequeno tutorial.

Um pequeno tutorial é exibido, vamos clicar no botão próximo e ao fim clicar em Done (Feito).

Um projeto inicial é exibido no centro da tela, nesse primeiro projeto aparece o nosso canvas, que é o nosso ambiente de desenvolvimento e é aqui que vamos desenvolver os nossos aplicativos.

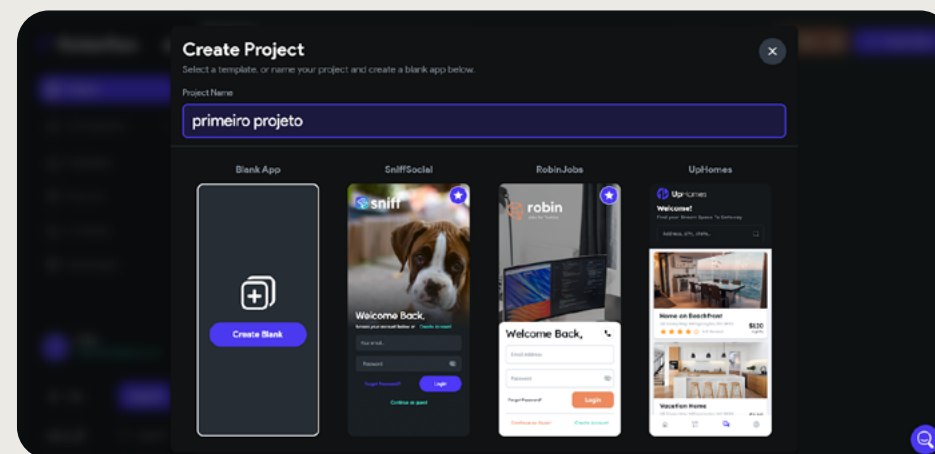
Conhecendo o ambiente de desenvolvimento:

Podemos criar apps simplesmente arrastando componentes como por exemplo: text (textos), row (linhas), column (colunas) para o nosso device e ver em tempo real as mudanças feitas. Ao clicar no componente adicionado, no canto superior direito estão as propriedades do componente, é aqui onde vamos editar e adicionar funções para criar um aplicativo bonito e funcional.

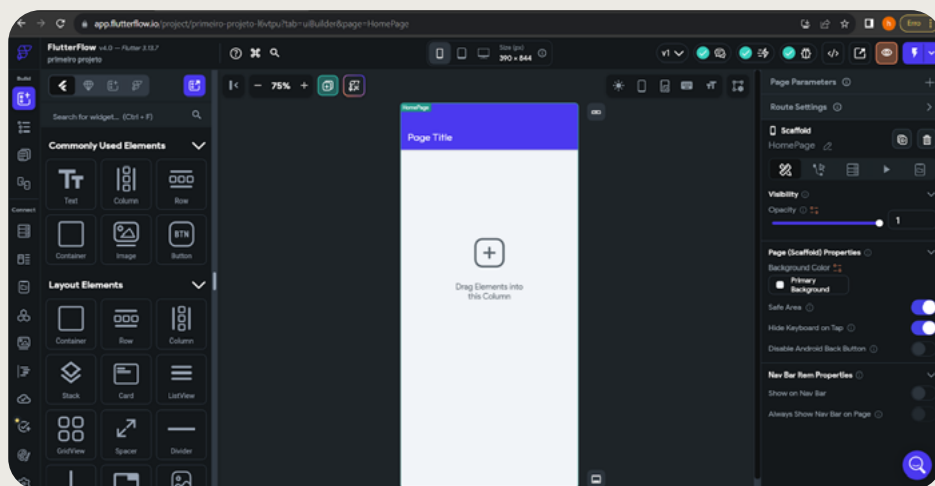
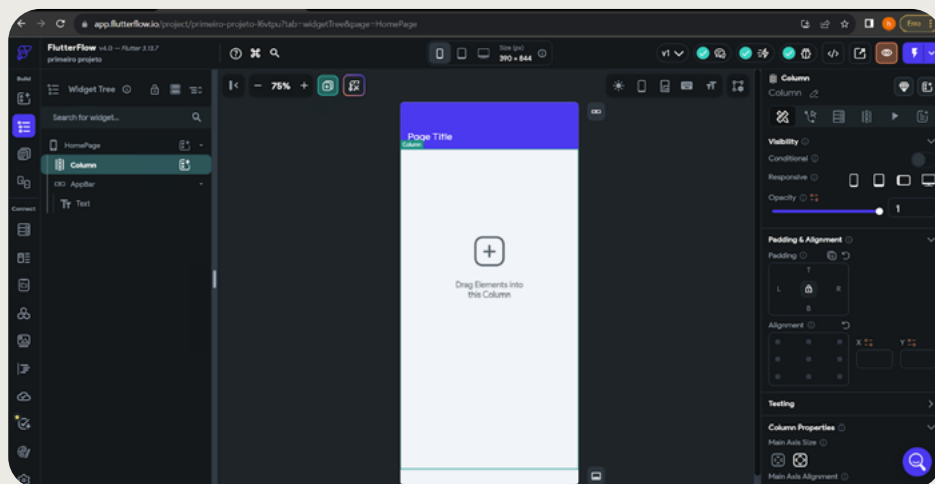
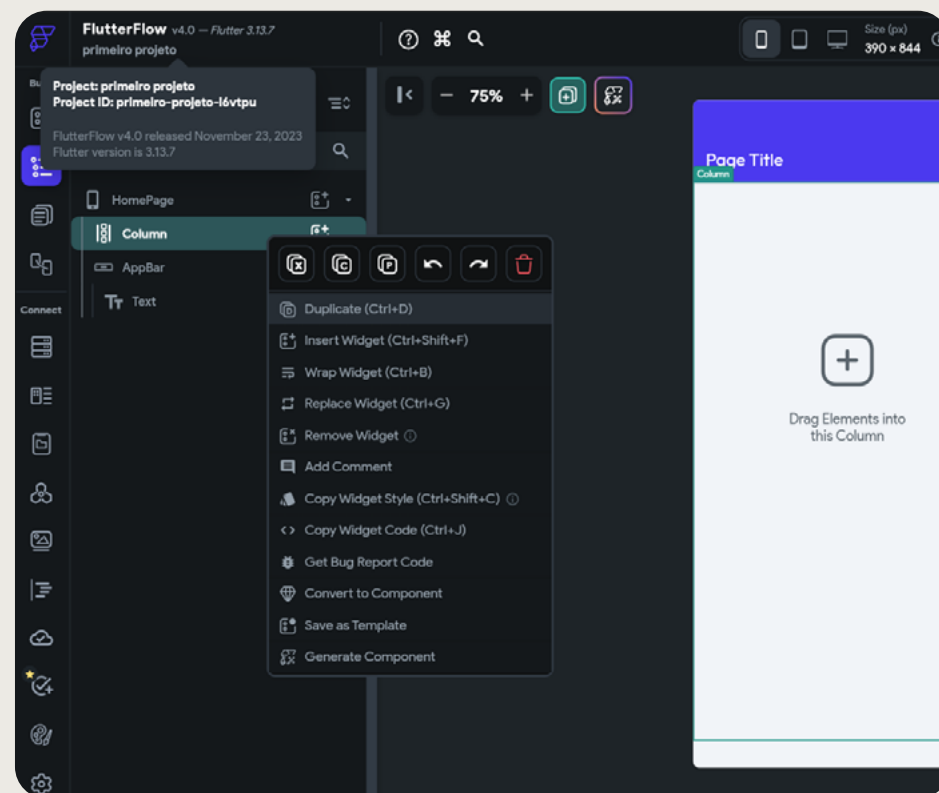
Hora de praticar...

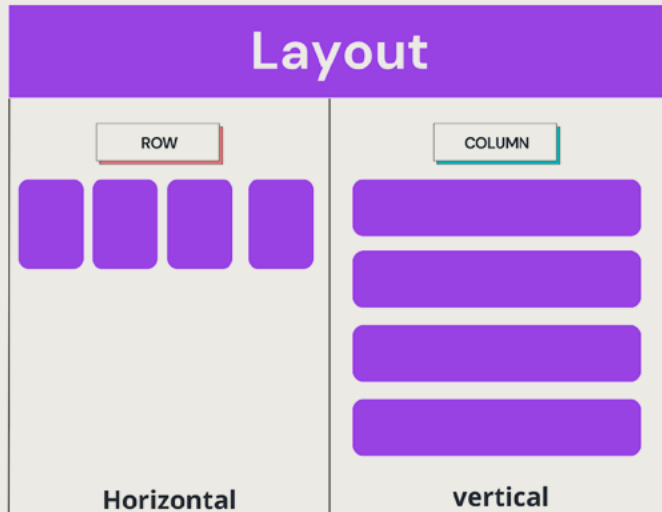
Clicando no ícone do FlutterFlow que se encontra no canto da tela, vamos para a tela inicial.

Nota:
Na tela inicial será exibida uma pesquisa, podemos responder ou simplesmente clicar no "X" para fechar.

Vamos criar um novo projeto:

Clique no botão **Create New** (Criar novo) para criar um novo projeto.

Explique aos alunos que o FlutterFlow oferece uma variedade de modelos de aplicativos pré-construídos que pode ser usado como ponto de partida para o seu projeto, porém vamos criar um projeto do 0.

Vamos dar um nome ao nosso projeto (primeiro projeto) e clicar em **Create Blank** (Criar em branco).

Nota: Ao criar um primeiro projeto, será exibida uma "boas-vindas". Clique em "Skip" para pular.

Chegamos ao nosso ambiente de desenvolvimento, veja que o FlutterFlow já trouxe uma tela com alguns componentes (Texto, AppBar, Coluna).

Vamos excluir os componentes para deixar um ambiente de desenvolvimento limpo, para isso clique em **Widget Tree**.

Foi exibida no lado direto a árvore de componentes adicionados a página, sendo a nossa HomePage, uma coluna, uma AppBar e um texto. Vamos excluir tudo que está abaixo da nossa HomePage, para isso basta clicar com o botão direto sobre o componente e clicar no ícone de lixeira no canto e o componente será deletado.

Linhas e colunas...

Vamos aprender agora sobre os dois componentes no FlutterFlow que são um dos mais usados no desenvolvimento de aplicativo, a linha e a coluna (Row e Column).

No FlutterFlow, assim como em outras linguagens de programação e frameworks, linha e coluna são elementos de layout que ajudam a organizar os widgets na interface do usuário. Aqui está a diferença entre linha e coluna no FlutterFlow:

Linha (Row):

Uma linha é um widget usado para organizar os elementos horizontalmente.

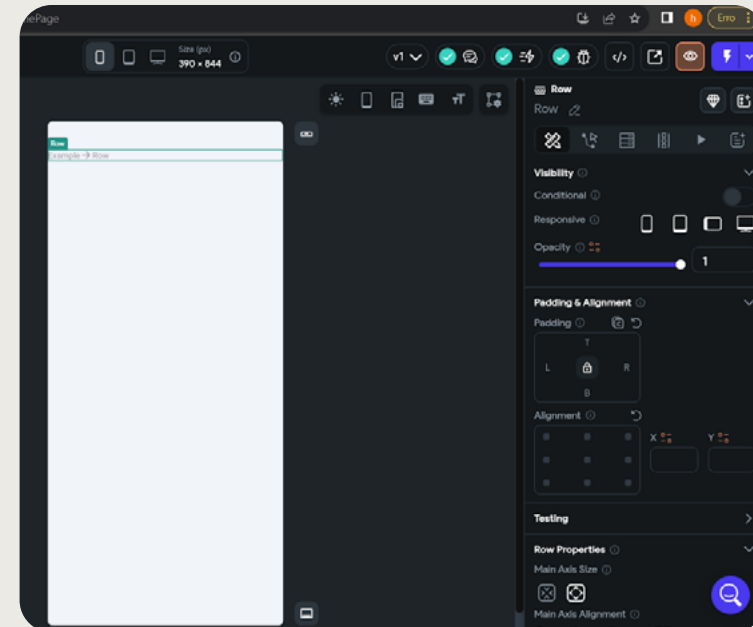
Os widgets são colocados lado a lado na horizontal dentro da linha.

Os elementos são dispostos em ordem, da esquerda para a direita.

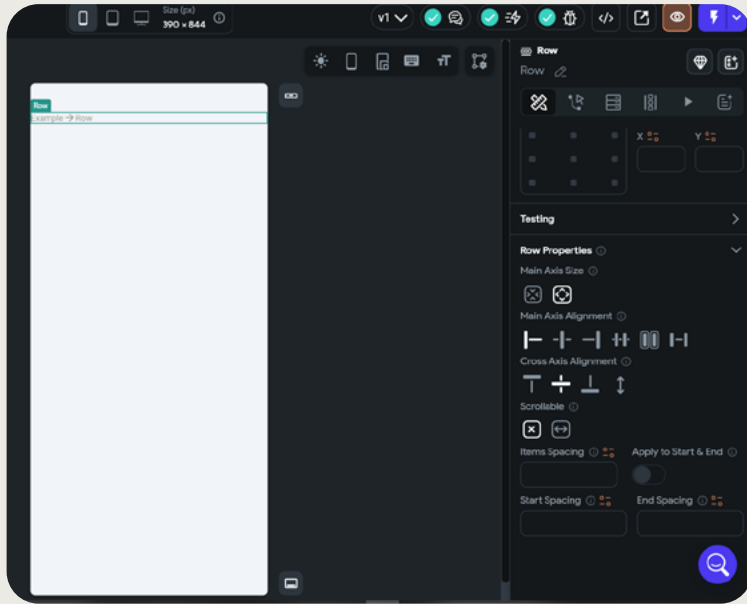
A largura de cada elemento é definida automaticamente com base no conteúdo.

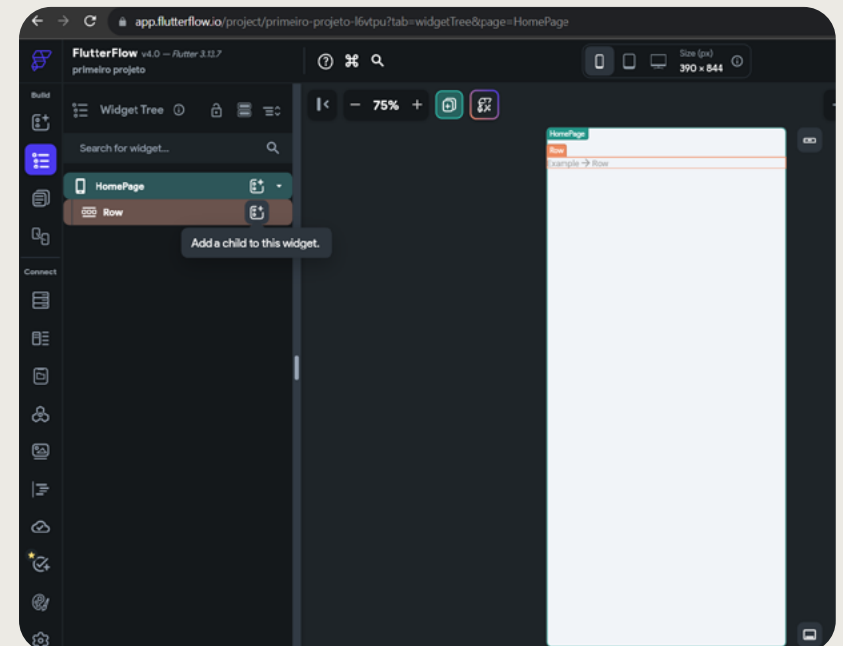
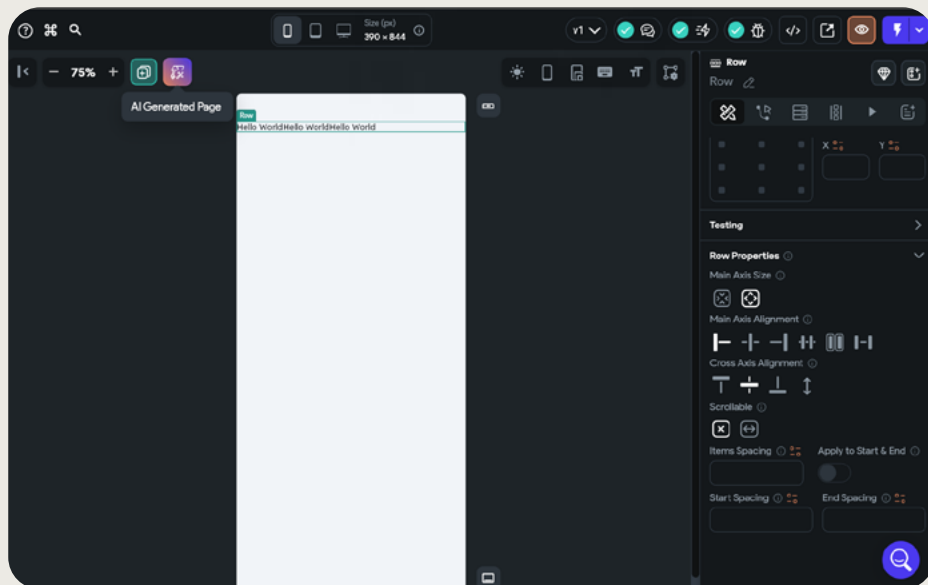
Se o espaço disponível for menor do que o necessário para acomodar todos os elementos, eles podem ser redimensionados ou cortados.

Você pode definir propriedades adicionais, como o espaçamento entre os elementos e o alinhamento vertical dentro da linha.

Vamos arrastar e soltar para a nossa HomePage o componente Row (linha).

Perceba que no ícone da row já são mostrados ícones um ao lado do outro, simbolizando, de fato, como uma linha se comporta.

Com a linha selecionada, podemos ver do lado esquerdo as propriedades dessa linha, é aqui onde faremos a estilização, como opacidade e o principal, mexer no alinhamento dessa linha.

Vamos descer até as últimas opções, as propriedades de alinhamento do componente.

Para uma maior compreensão, vamos adicionar 3 textos dentro da nossa Row, podemos fazer isso de duas maneiras, sendo a primeira arrastando o componente Text para dentro do componente Row na nossa HomePage (repetindo o processo 3 vezes).

E a segunda através da nossa árvore de componentes na widget Tree e clicando no ícone ao lado da Row (Add a child to this widget.) e selecionando o componente Text (repetir o processo 3 vezes).

Com os textos adicionados vamos ver as propriedades de alinhamento da nossa Row.

Clique na linha e vá para as propriedades de alinhamento.

Main Axis Size:

1ª opção: Use a quantidade mínima de tamanho no eixo principal.

2ª opção: Use a quantidade máxima de tamanho no eixo principal. Expande para preencher o espaço disponível.

Main Axis Alignment:

1ª opção: Começar. Coloque os elementos filhos o mais próximo possível do início.

2ª opção: Centro. Coloque os elementos filhos o mais próximo possível do meio.

3ª opção: Fim. Coloque os elementos filhos o mais próximo possível do final.

4ª opção: Espaço uniformemente. Espaçar uniformemente os elementos filhos.

5ª opção: Espaço ao redor. Coloque o espaço livre uniformemente entre os elementos filhos com algum espaço extra no início e no final.

6ª opção: Espaço entre. Coloque o espaço livre uniformemente entre os elementos filhos.

Cross Axis Alignment (para ver essas opções na prática, adicione um container no início dos textos):

1ª opção: Começar. Coloque os elementos filhos o mais próximo possível do início.

2ª opção: Centro. Coloque os elementos filhos o mais próximo possível do meio.

3ª opção: Fim. Coloque os elementos filhos o mais próximo possível do final.

4ª opção: Esticar. Faz com que os filhos preencham o eixo transversal (verticalmente).

Scrollable:

1ª opção: Não permite rolagem de tela.

2ª opção: Permite rolagem de tela.

Temos outras propriedades que veremos ao decorrer do curso.

Coluna (Column):

Uma coluna é um widget usado para organizar os elementos verticalmente.

Os widgets são empilhados verticalmente dentro da coluna.

Os elementos são dispostos em ordem, de cima para baixo.

A altura de cada elemento é definida automaticamente com base no conteúdo.

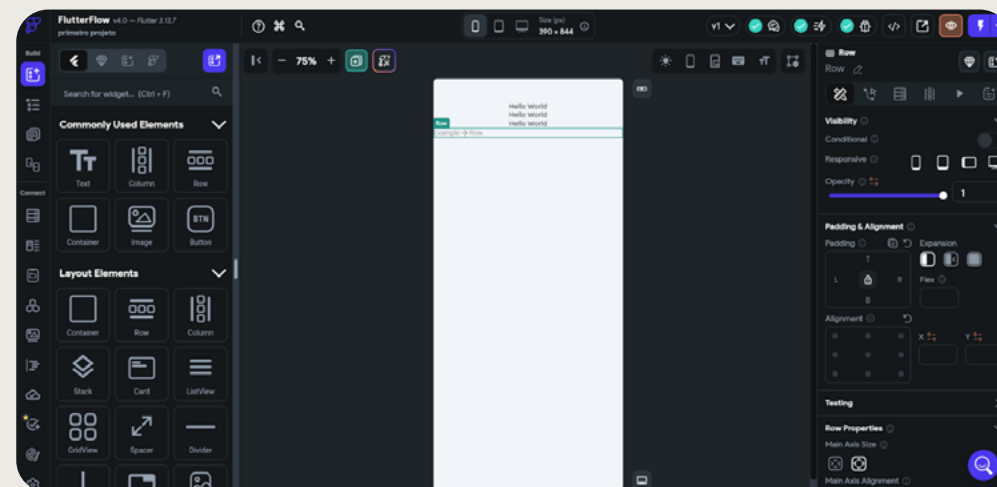
Se o espaço disponível for menor do que o necessário para acomodar todos os elementos, eles podem ser redimensionados ou cortados.

Você pode definir propriedades adicionais, como o espaçamento entre os elementos e o alinhamento horizontal dentro da coluna.

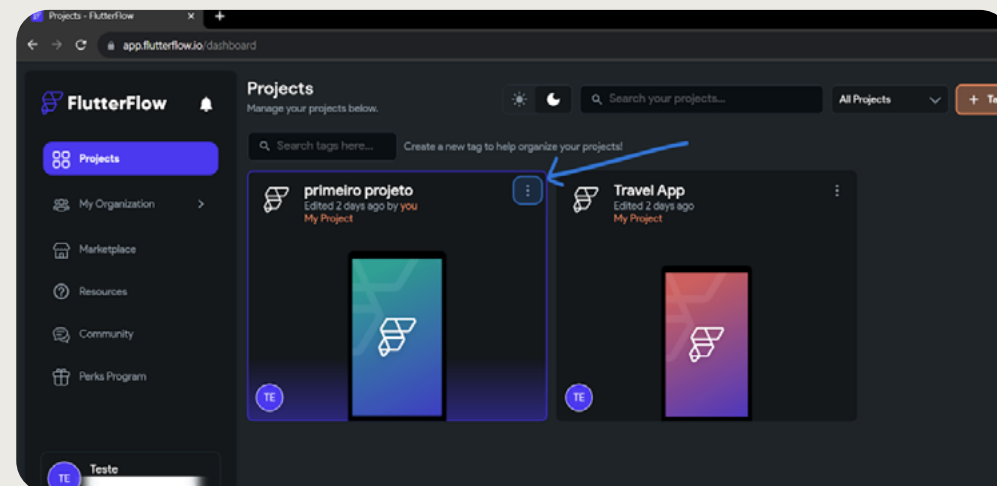
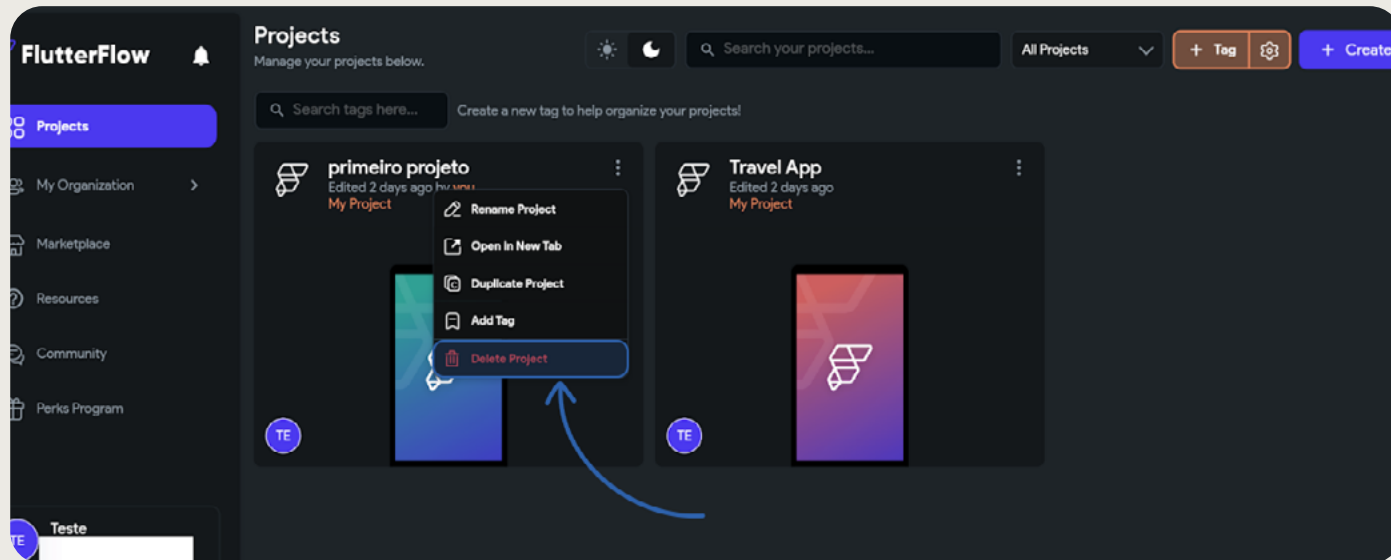
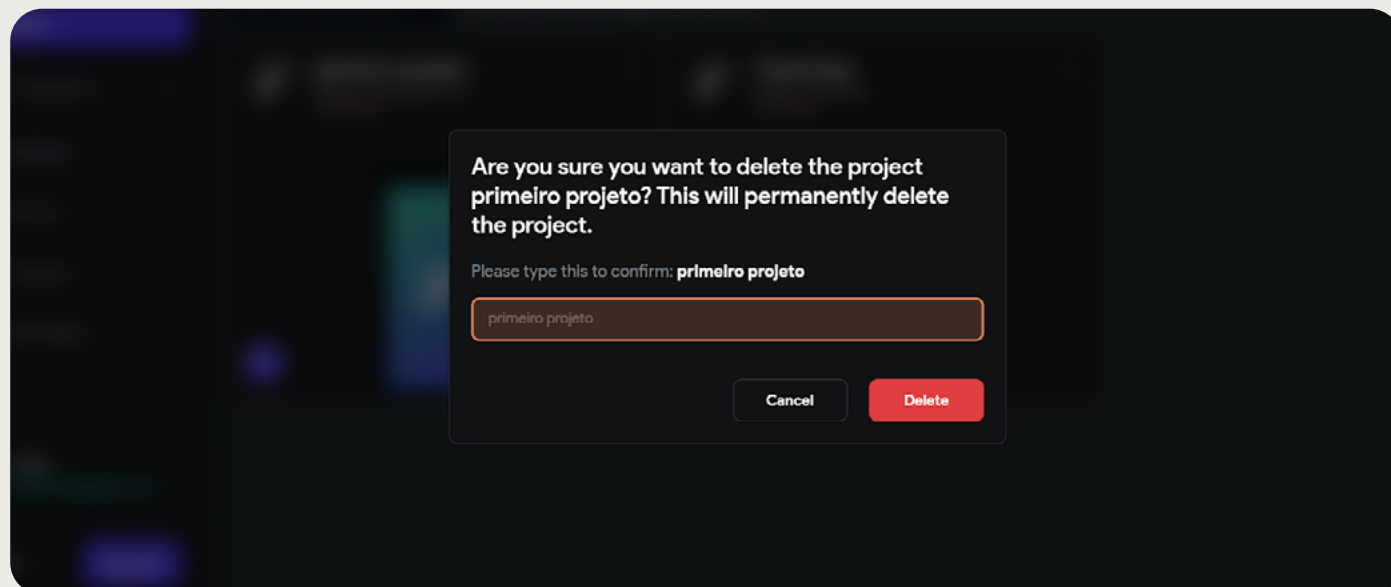
Em resumo, a principal diferença entre linha e coluna no FlutterFlow é a orientação dos elementos: linha organiza horizontalmente e coluna organiza verticalmente. A escolha entre linha e coluna depende de como você deseja posicionar os elementos na interface do usuário do seu aplicativo FlutterFlow.

Na prática...

Para esse exemplo prático, instrua os alunos a excluir a linha com os textos. Com o ambiente de desenvolvimento limpo, arraste para a HomePage o componente Column.

Vamos fazer a mesma coisa que fizemos na linha e adicionar 3 textos, veja que dessa vez os textos ficaram um abaixo do outro, é dessa forma que esse componente se comporta, podendo adicionar quantos componentes quiser, até mesmo uma linha dentro de uma coluna e assim podendo fazer combinações incríveis!

Excluindo um projeto:

Vamos aprender a deletar um projeto, para isso navegue até a tela inicial do FlutterFlow clicando no ícone conforme já foi ensinado.

No projeto que deseja deletar clique no menu de opções (três pontinhos).

Aparecerão algumas opções, clique em **Delete Project** (deletar projeto).

Aparecerá uma mensagem de confirmação, digite o nome do projeto, clique em delete e ele será deletado com sucesso.

AULA 02

FlutterFlow

Materiais necessários:
Notebook, acesso à internet.

Duração:
2 horas

Objetivo:
Conhecer o FlutterFlow.

ENGAJAR

Vamos criar um novo, conforme também já foi ensinado, crie um projeto com o nome "Lista de alunos", através deste aplicativo, aprenderemos **conceitos** e **funções** muito importantes.

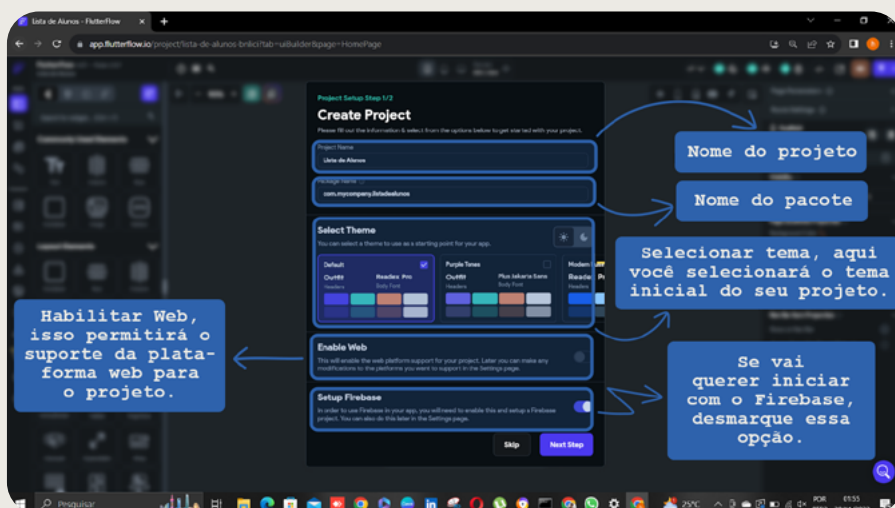
Ao criar um projeto, aparecerá uma tela com algumas informações:

Nome do projeto

Nome do pacote

Selecionar tema, aqui você selecionará o tema inicial do seu projeto.

Habilitar Web, isso permitirá o suporte da plataforma web para o projeto.

Se vai querer iniciar com o Firebase, desmarque essa opção.

Desmarque a opção **Setup Firebase** e clique em **Start Building**.

Imagens...

Delete os componentes que vêm de padrão no nosso device, deixando um ambiente de desenvolvimento limpo.

Vamos criar o início de uma lista com os nomes dos alunos.

Explique aos alunos o que vamos criar e pergunte qual componente usaremos para fazer uma lista (por um componente abaixo do outro).

Resposta: Column (Coluna).

Arraste o componente coluna até o device, ou clique no botão Add Child na HomePage e procure o componente column.

Com uma coluna adicionada, precisamos fazer agora a nossa AppBar, contendo uma imagem do aluno, nome e um menu itens.

Novamente pergunte aos alunos qual componente é utilizado para alinhar um item ao lado do outro.

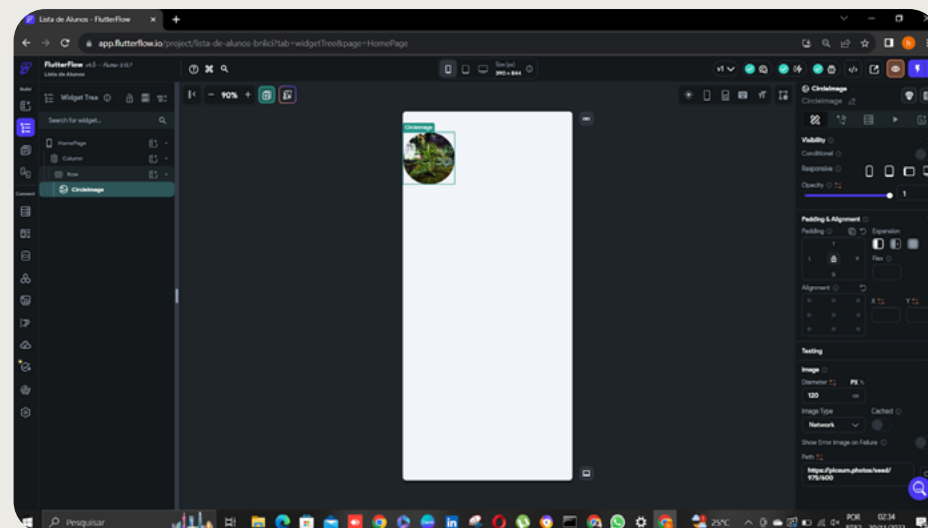
Resposta: Row (Linha).

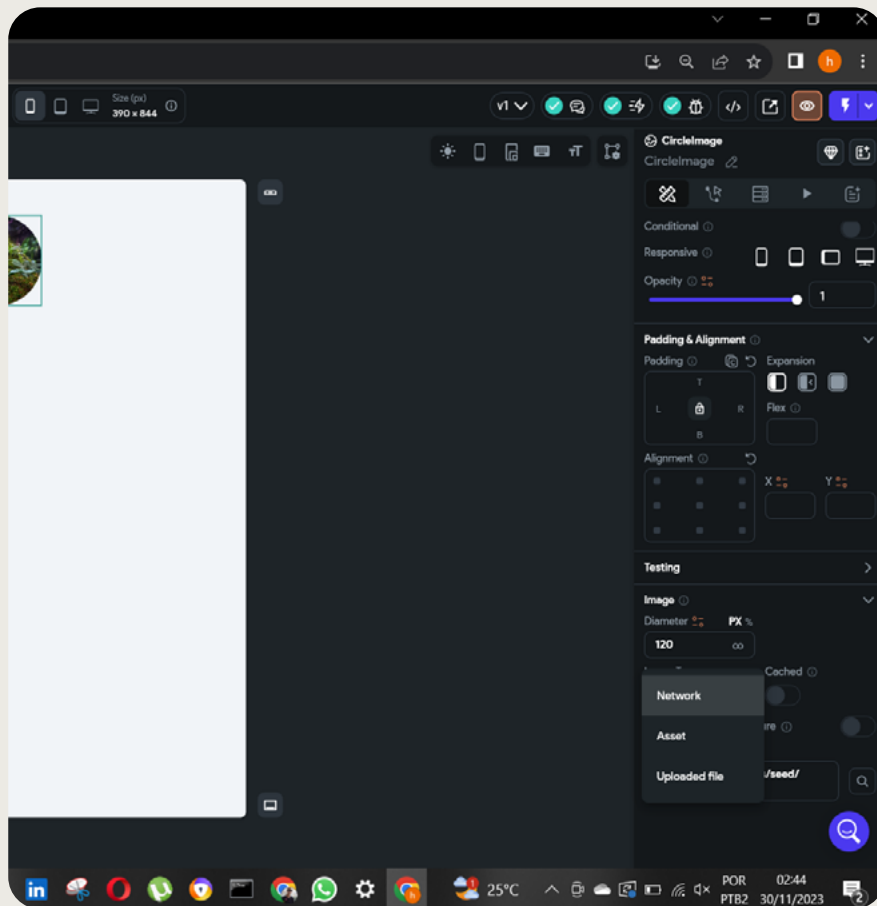
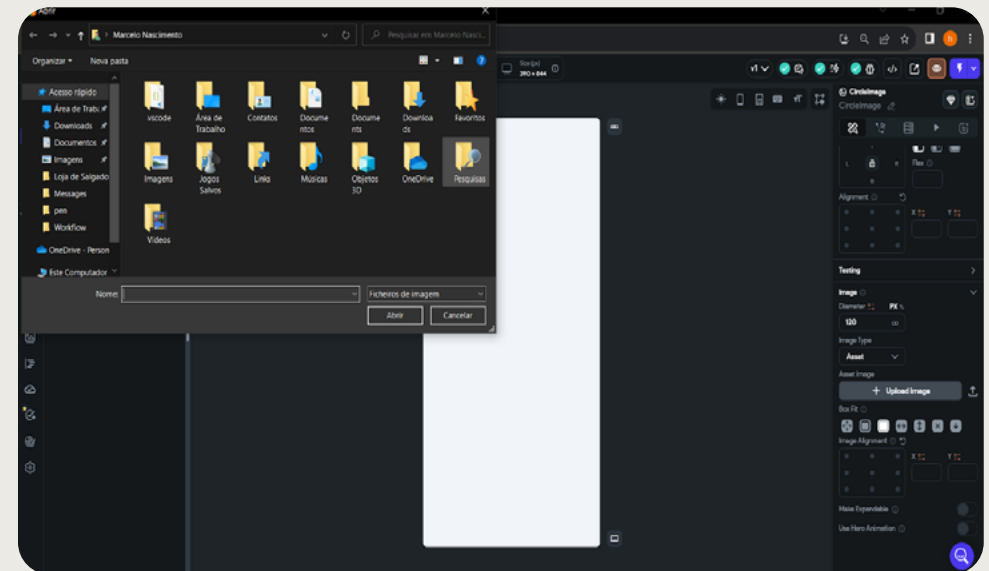
Adicione uma linha dentro da coluna, perceba que estamos unindo dois componentes para formar o nosso layout desejável.

Imagens...

Vamos adicionar uma imagem circular a nossa AppBar e explorar as propriedades dela.

Dentro da linha adicione o componente Circleimage (imagem circular), use a barra de pesquisa para encontrar.

Perceba que uma imagem foi adicionada e automaticamente as propriedades dessa imagem aparecem no lado direito do canvas.

Na opção **image Type**, será o local de onde a imagem será adicionada, por padrão o FlutterFlow traz uma imagem da web, o link da imagem está na opção abaixo **"Path"**.

Vamos adicionar uma imagem local da própria máquina, para isso, clique na opção **image Type** e clique em asetss.

A imagem que estava desaparecerá. **Clique em Update image**, o explorador de arquivos será aberto. Escolha a imagem desejada e clique em abrir.

No componente imagem, há mais algumas opções de estilização, como dimensão da imagem (o tamanho) e o Box Fit.

O que é o Box Fit?

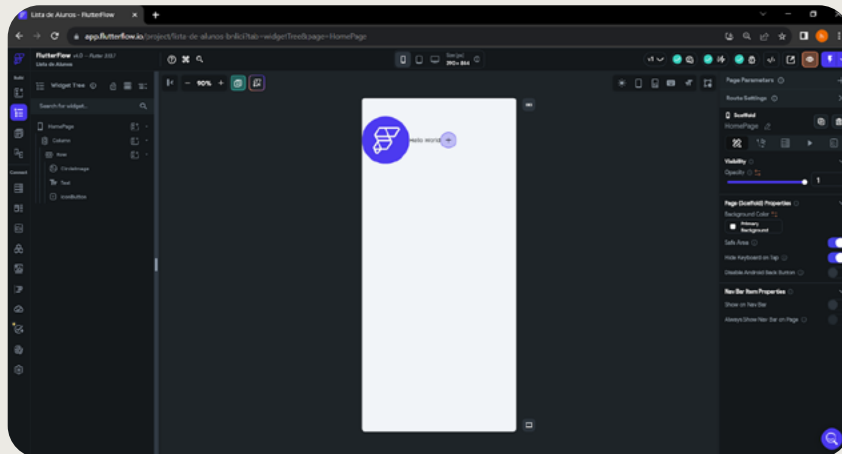
Box Fit é uma propriedade usada para definir como uma imagem deve ser ajustada dentro do espaço que lhe foi designado. No contexto do FlutterFlow, o Box Fit pode ser uma opção disponível para personalizar o layout e o comportamento das imagens em seu aplicativo FlutterFlow. Ao selecionar o Box Fit, você pode controlar se a imagem deve ser redimensionada para preencher o espaço disponível (preenchimento), ajustada para caber dentro do espaço sem distorção (contido) ou alinhada de forma específica dentro do espaço designado (alinhado).

Essa funcionalidade pode ser útil para garantir que as imagens em seu aplicativo FlutterFlow sejam exibidas da maneira desejada, independentemente do tamanho ou proporção original das imagens.

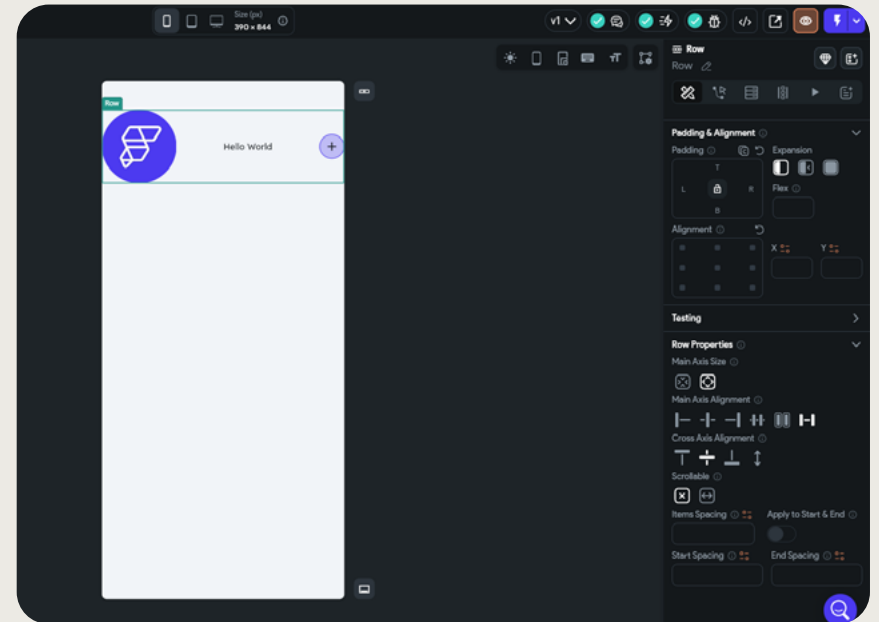
INVESTIGAR

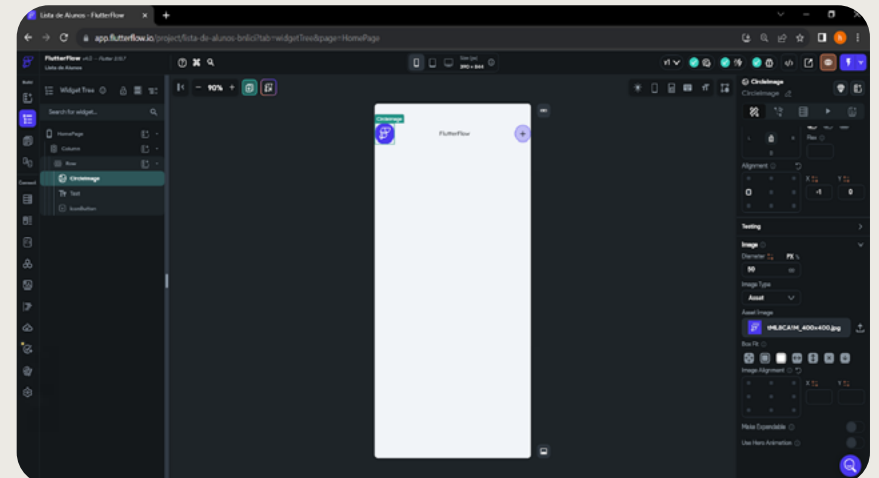
Instrua os alunos a verem como cada opção se comporta.

Seguindo o raciocínio de uma **AppBar**, vamos adicionar um texto e um **IconButton** dentro da **Row** (linha).

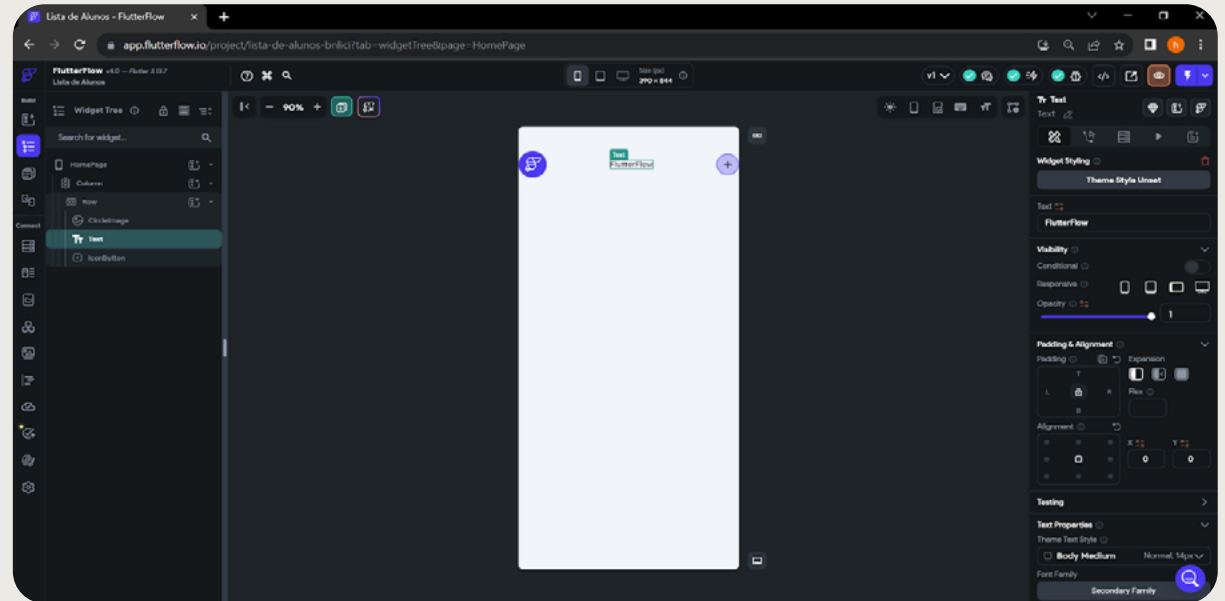
Perceba que os componentes ficaram bem juntos um do outro.

Fixando o que já sabe...

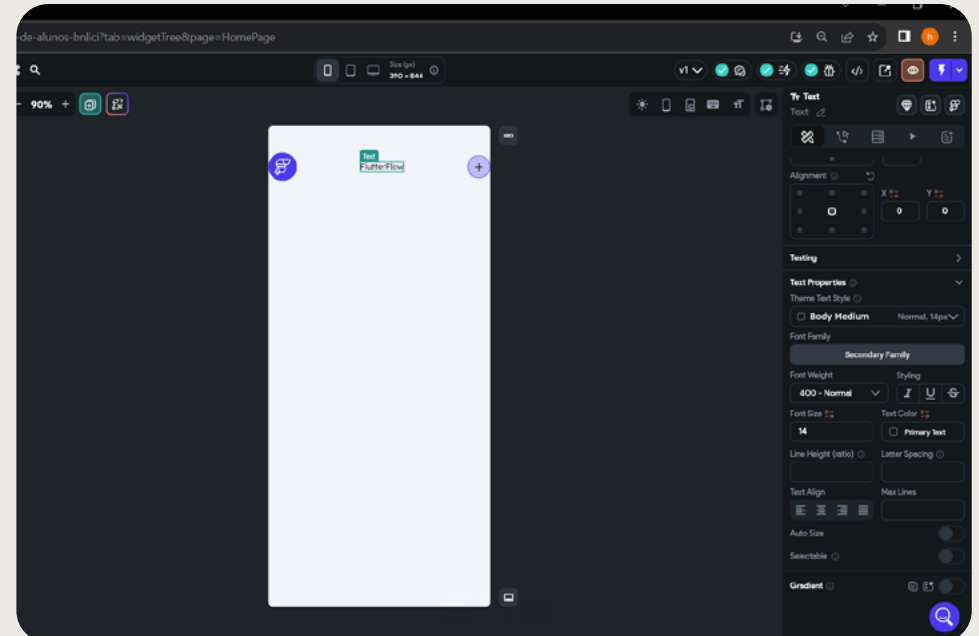
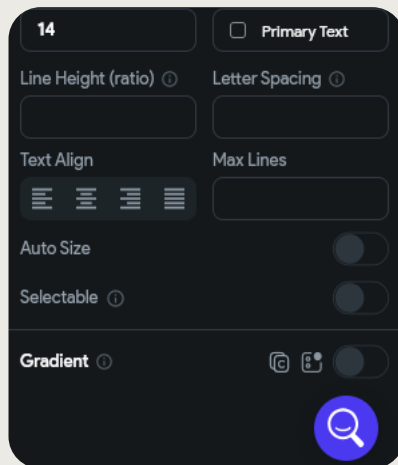
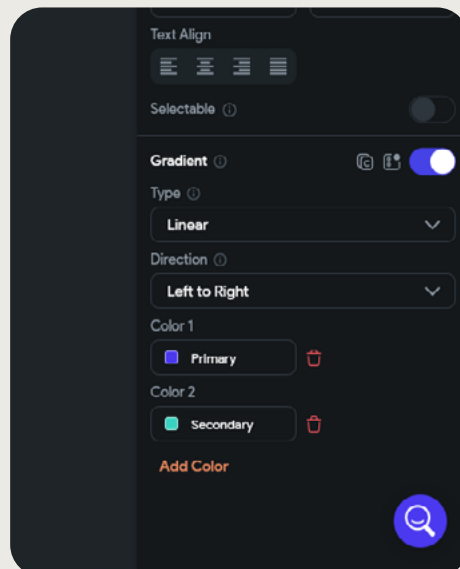
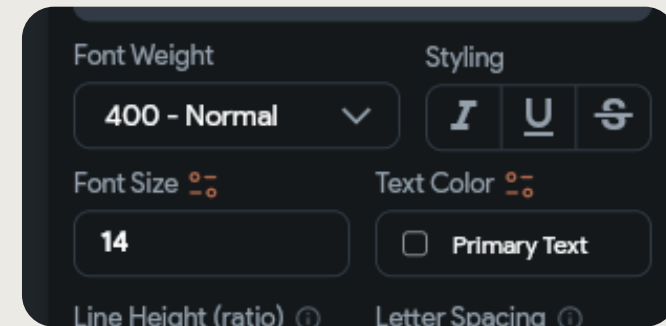
Pergunte aos alunos como podemos fazer essa separação de forma adequada, deixando a imagem no canto, texto no meio e ícone no final.

Resposta: Clicando na linha e selecionando a última opção na Main Axis Alignment (a opção Space Between).

Veja que os componentes da linha foram alinhados conforme esperado.

Agora precisamos diminuir a imagem deixando com uma dimensão de 50px.

Textos...

Clicando no componente texto no nosso device podemos ver todas as opções de modificações que ele nos traz nas propriedades do mesmo.

Vamos modificar o texto que está escrito por padrão, para isso vá até a opção Text e altere o texto que está escrito pelo nome do aluno, perceba que o texto foi modificado em tempo real.

Podemos fazer muitas outras modificações alterando o que for necessário para deixar com o layout desejável, como por exemplo:

Modificar a fonte do texto - Em Font family podemos escolher entre várias fontes do FlutterFlow ou customizar uma nova.

Ainda falando sobre as propriedades dos textos podemos fazer diversas modificações, como o **Text Align**.

Pode adicionar um Gradiente no texto marcando a opção Gradiente e selecionando as cores.

Fazer a formatação do texto, alterar tamanho da fonte e cor.

Dentre outras coisas.

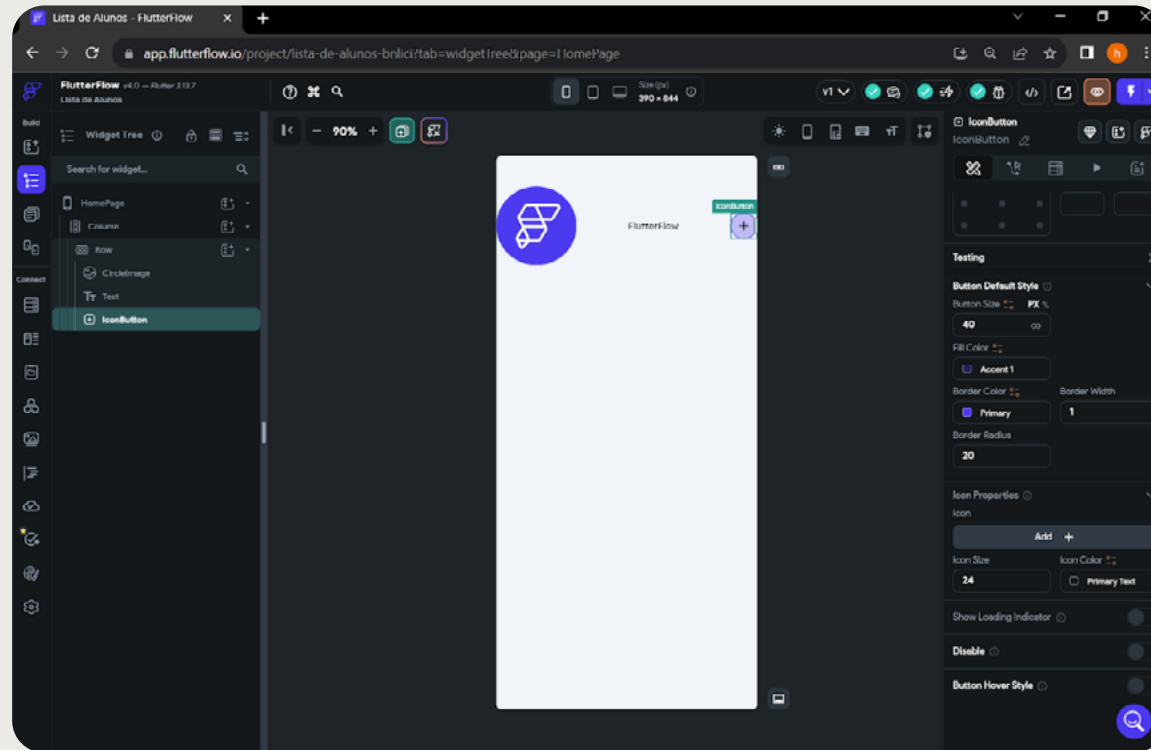
IconButton...

O que é IconButton?

O IconButton é um componente que representa um botão com um ícone. Este componente é usado para criar botões que contêm ícones e que podem ser acionados pelo usuário, geralmente para realizar uma ação específica dentro do aplicativo.

Ao adicionar um IconButton em seu projeto FlutterFlow, você pode escolher um ícone a partir de uma variedade de opções disponíveis, como ícones pré-definidos no Flutter ou ícones personalizados. Além disso, você pode atribuir uma ação ou função ao IconButton para que algo aconteça quando o usuário interage com ele, como navegar para outra tela, enviar um formulário ou executar qualquer outra ação desejada.

Em resumo, o IconButton é um componente interativo que combina um ícone visualmente reconhecível com a capacidade de acionar uma ação quando pressionado, tornando-o uma ferramenta útil para a criação de interfaces de usuário em aplicativos FlutterFlow.

Vamos clicar no componente IconButton e ver as propriedades dele.

Como se trata de um ícone dentro de um botão, temos as propriedades do botão e as propriedades do ícone dentro do botão.

No botão podemos:

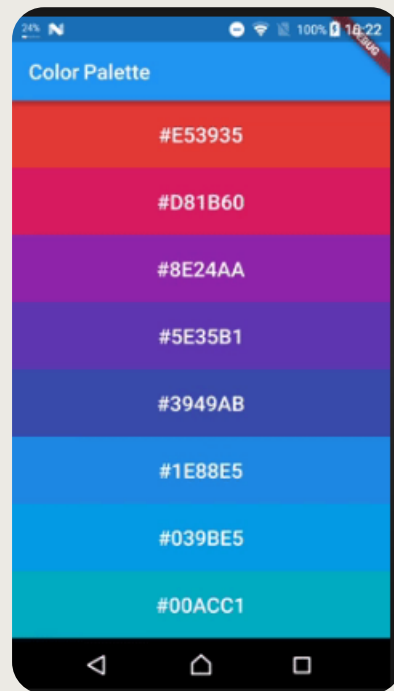
- Alterar o tamanho;
- Cor do botão;
- Cor da borda;
- Tamanho da borda;
- Arredondamento da borda.

Já nas propriedades do ícone podemos:

- Alterar o ícone que está sendo exibido;
- O tamanho do ícone;
- A cor etc.

Perceba que os componentes ficaram bem juntos uns dos outros.

AGIR

Peça para os alunos fazerem um exercício utilizando os conhecimentos obtidos durante as aulas.

AULA 03

FlutterFlow

Materiais necessários:
Notebook, acesso à internet.

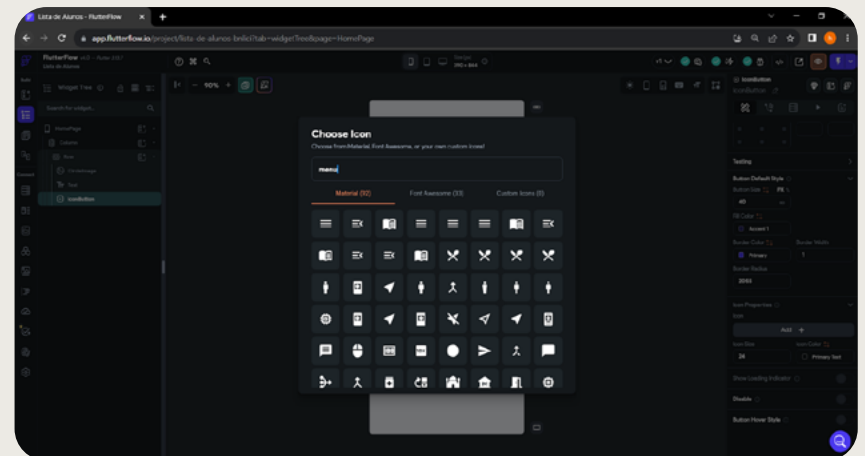
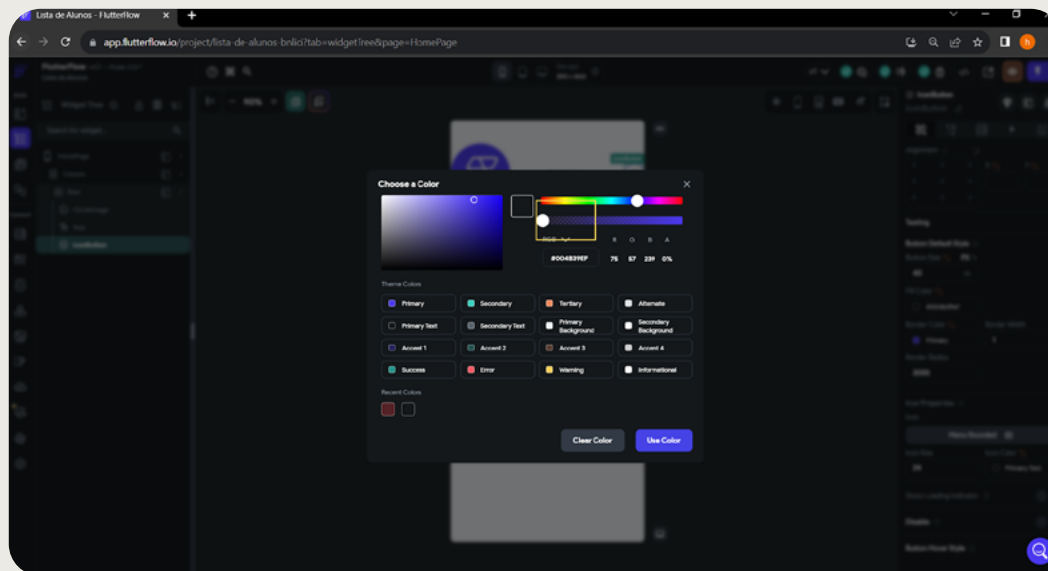
Duração:
2 horas

Objetivo:
Conhecer o FlutterFlow.

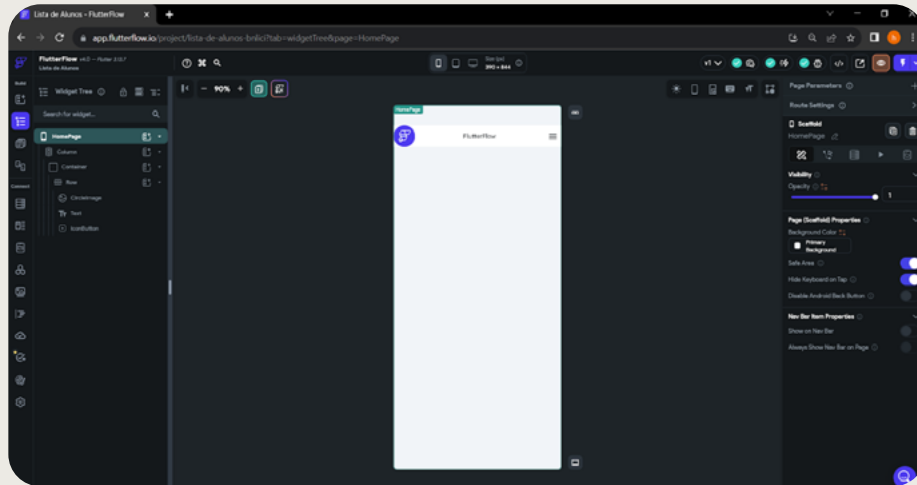
ENGAJAR

Continuando a aula anterior...

Vamos alterar o ícone que está dentro do botão, para isso clique Ícone e serão abertas várias opções de ícones para escolher.

Escolha o ícone menu, para isso digite na barra de pesquisa o nome menu e escolha uma opção desejável.

Nota: é importante ressaltar que todos os ícones estão em inglês.

Com o ícone alterado vamos agora deixar o fundo dele transparente.

No **fill color** e no **border color** arraste a cor para o transparente e clique em use Color.

Até o momento a nossa AppBar está assim:

Porém ainda não está muito atraente de se ver, portanto vamos aprender um novo componente.

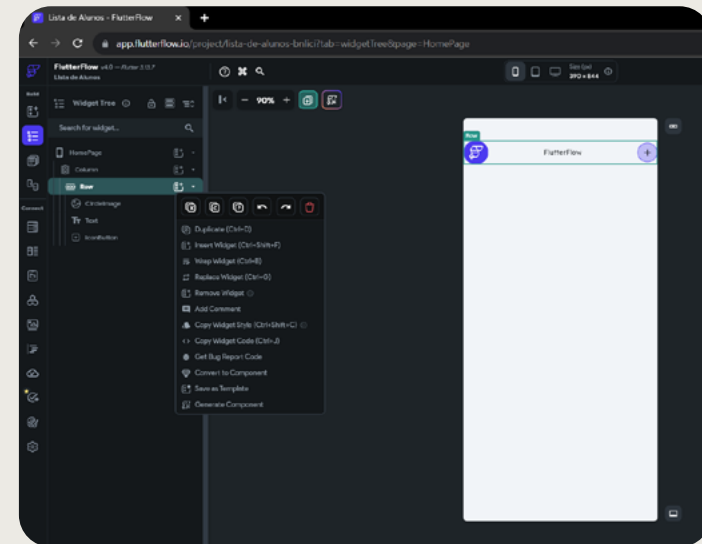
Container...

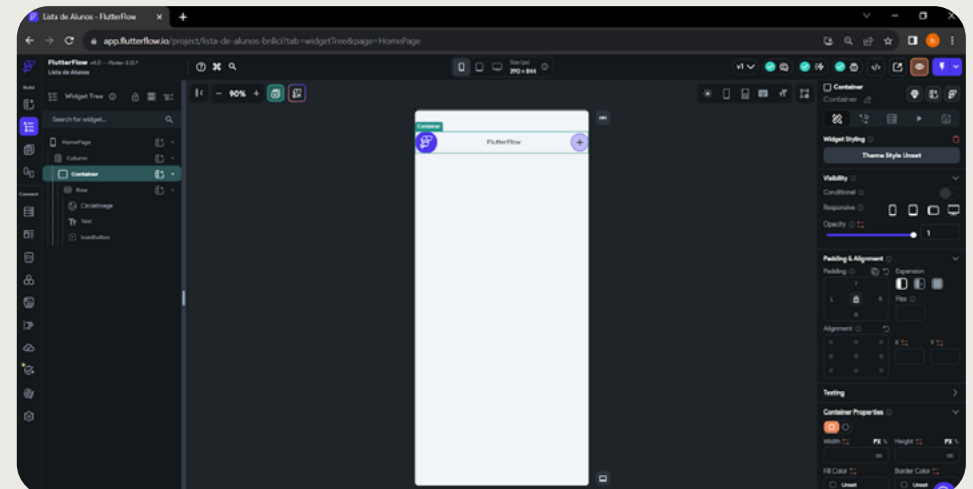
O que é um container?

Um Container é um componente utilizado para definir a aparência, layout e comportamento de um elemento visual em um aplicativo Flutter. Ele é frequentemente utilizado para envolver outros elementos, como texto, imagens, botões e outros componentes visuais, permitindo que sejam estilizados e organizados de acordo com as necessidades do design da interface do usuário.

Ao adicionar um Container em seu projeto FlutterFlow, você pode configurar uma variedade de propriedades, como cor de fundo, margens, preenchimento, bordas e sombras, entre outras opções. Isso permite que você controle o espaçamento, a aparência e a interação do elemento visual que está dentro do Container.

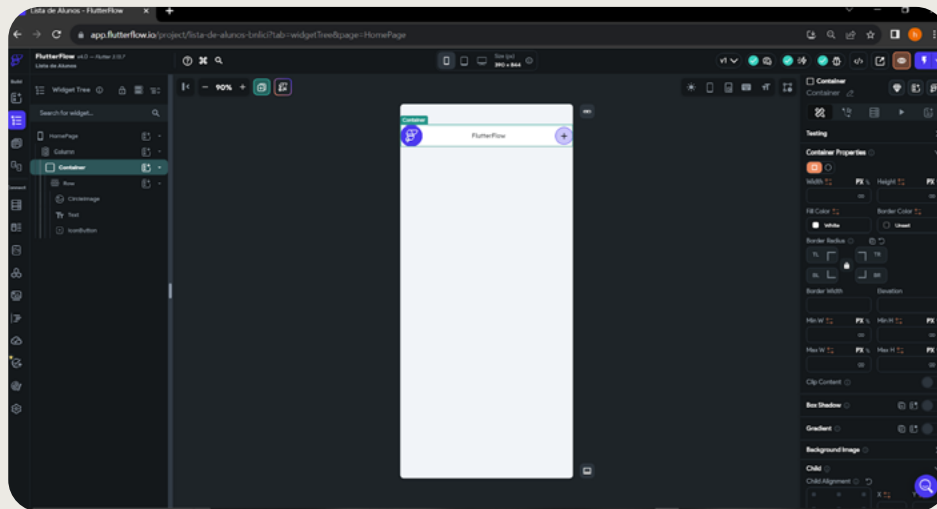
Em resumo, o Container no FlutterFlow é um elemento fundamental para a criação de layouts e designs de interface de usuário, pois oferece uma maneira flexível e poderosa de estimular e organizar os elementos visuais em seu aplicativo FlutterFlow.

Vamos adicionar um **container** dentro da nossa **row**. Para isso, clique com o botão direito do mouse sobre a linha e escolha a opção **Wrap Widget**.

Aparecerão componentes que você pode usar, escolha um **container**.

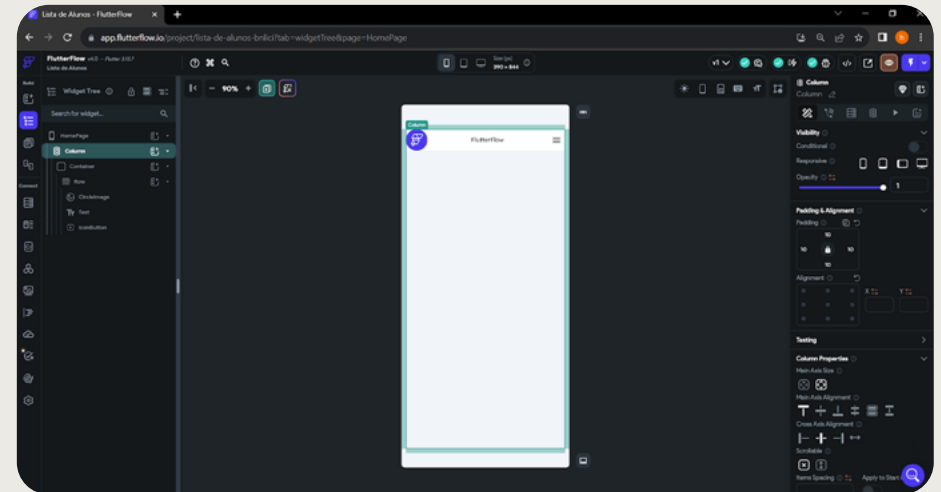
Perceba que acima da **linha** apareceu o componente **container**, clique sobre ele.

Nas propriedades ao lado podemos fazer diversas alterações, tais como:

- Adicionar uma cor;
- Adicionar uma imagem de fundo;
- Um gradiente;
- Sombras;
- Tamanho da borda;
- Cor da borda;
- Podemos deixar a imagem circular, entre outras coisas.

Vamos alterar a cor do nosso container.

Para isso vá até a **Fill color** e selecione a cor White (branco).

Perceba que os componentes estão muito perto da margem da tela, vamos dar um espaço para um melhor layout.

Para isso clique na **coluna**.

Vamos na propriedade **Padding & Alignment** e dê um padding de 10 em todas as extremidades.

Perceba que agora toda a coluna apresenta um espaço nas bordas.

Vamos prosseguir com o desenvolvimento da nossa lista.

Vamos aprender um novo componente.

Card...

O que é um card?

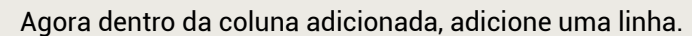
Um Card é um componente utilizado para exibir conteúdo de forma organizada e estruturada em um aplicativo Flutter. Os Cards são frequentemente utilizados para apresentar informações importantes de maneira visualmente atraente, como por exemplo, em listas de itens ou em seções de destaque.

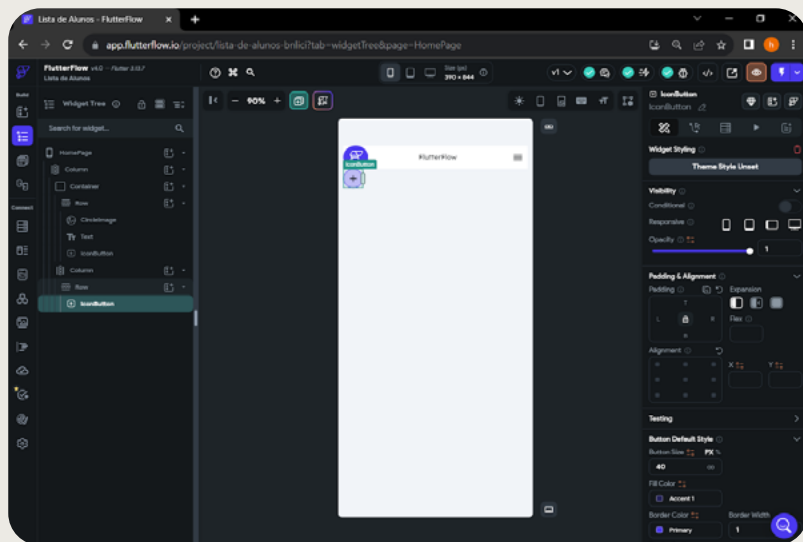
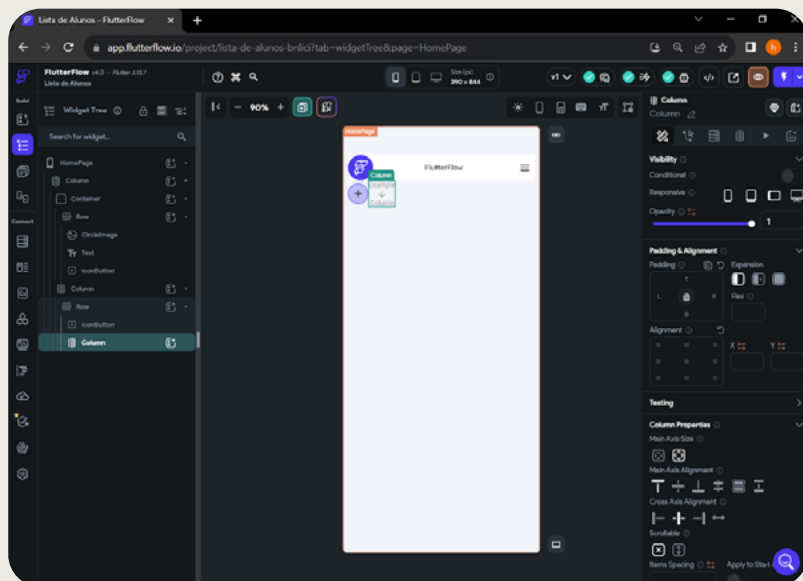
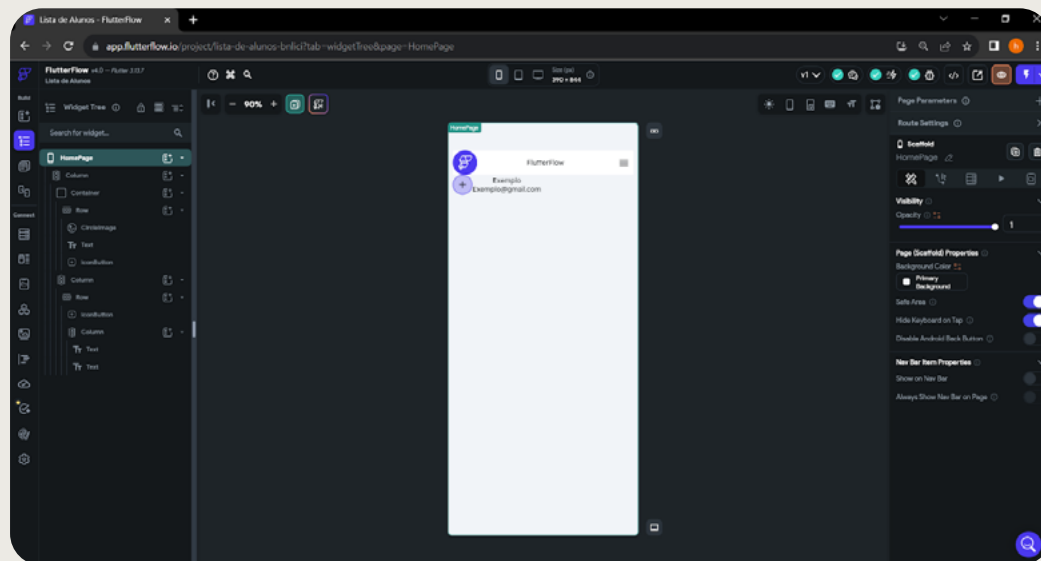
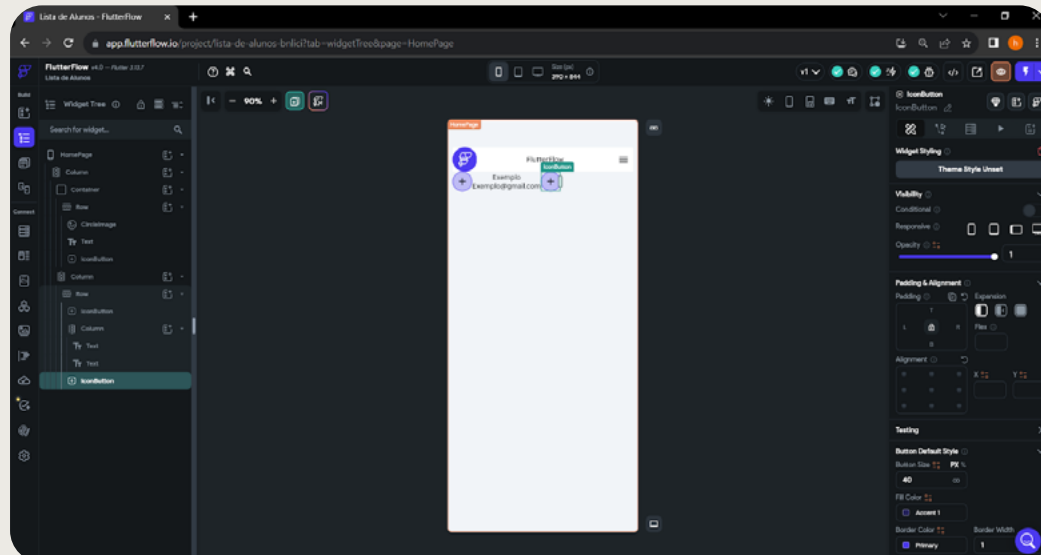
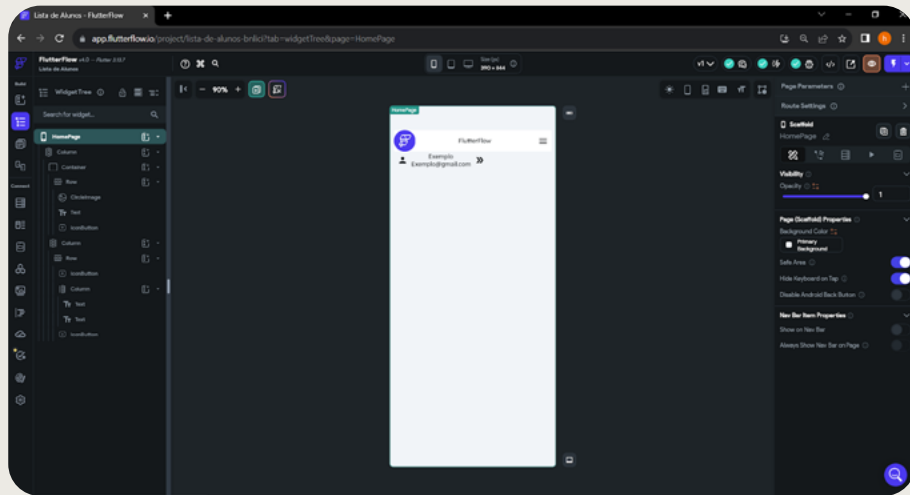
Ao adicionar um Card em seu projeto FlutterFlow, você pode configurar uma variedade de propriedades, como cor de fundo, sombra, bordas, espaçamento interno e conteúdo, permitindo que você personalize a aparência e o comportamento do Card de acordo com as necessidades do seu design de interface de usuário.

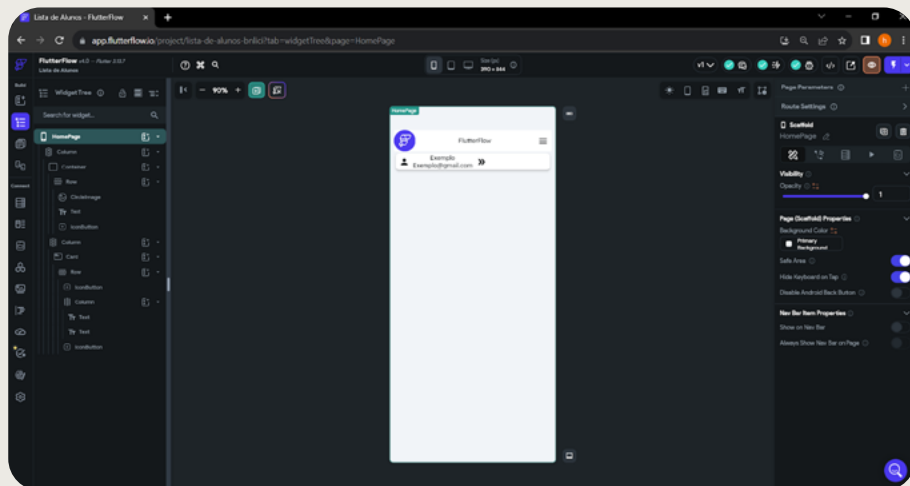
Os Cards são frequentemente utilizados para criar visualmente agrupamentos de conteúdo, como artigos, produtos, mensagens ou qualquer outra informação que precise ser destacada em um layout. Eles oferecem uma maneira eficaz de organizar e apresentar informações de forma consistente e atraente.

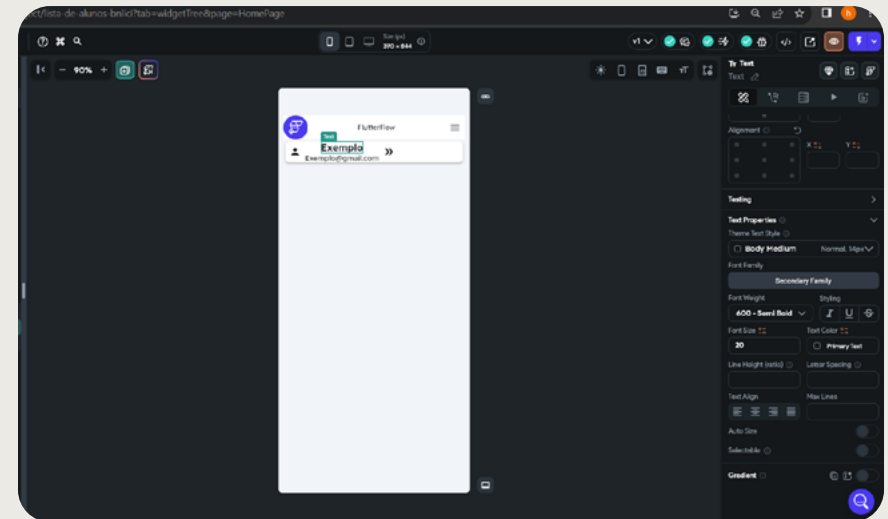
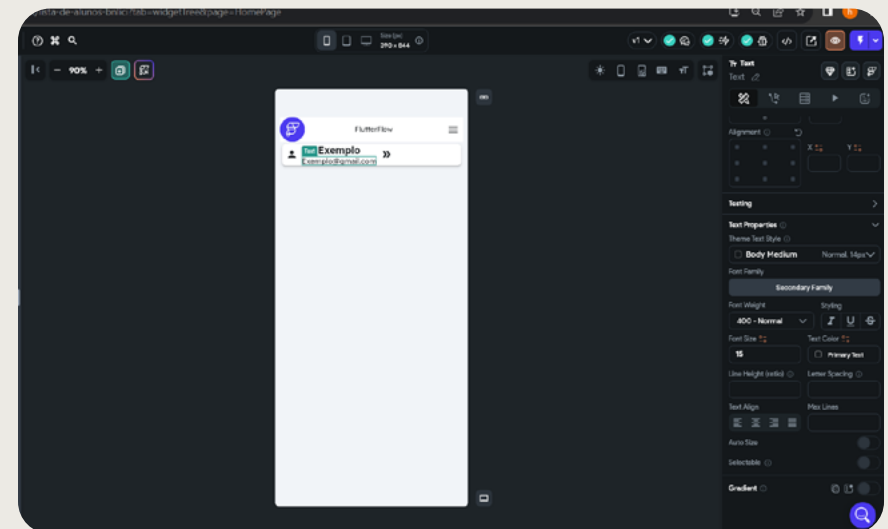
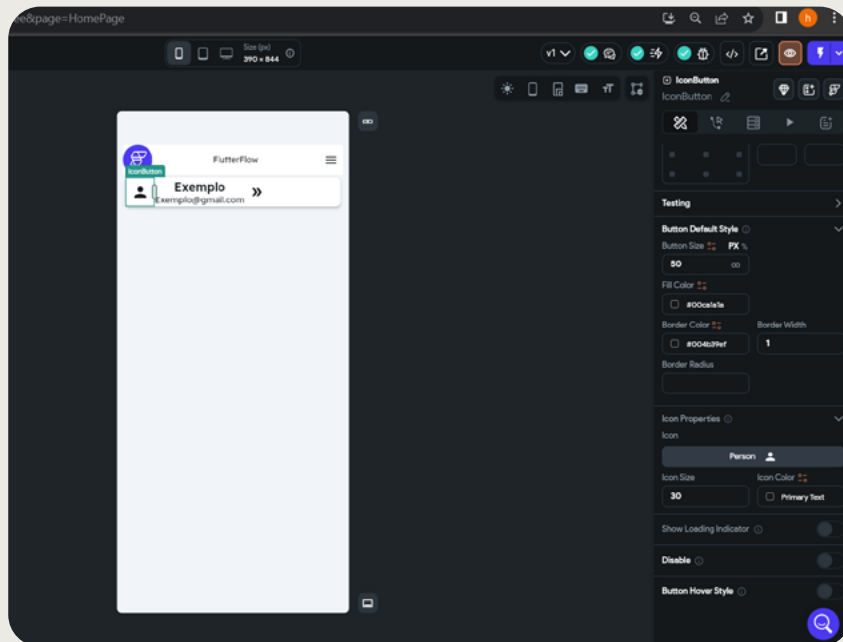
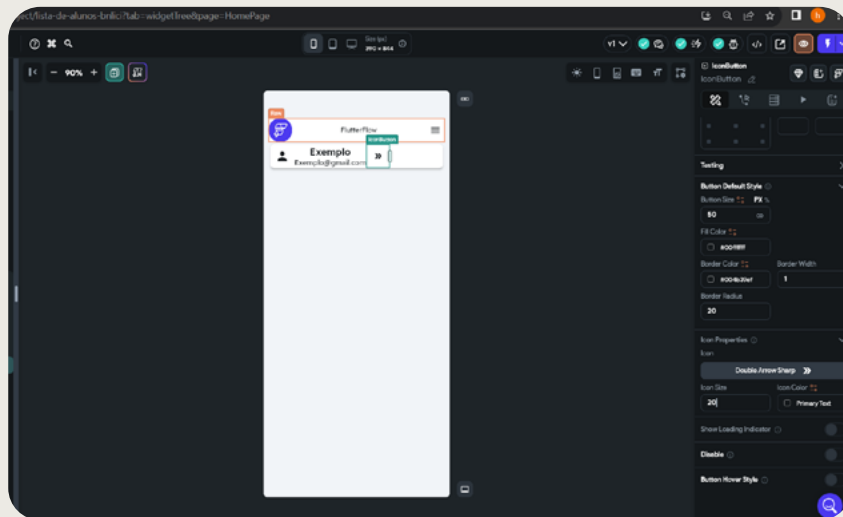
Em resumo, no FlutterFlow, o Card é um componente versátil que permite a exibição de conteúdo de forma estruturada e visualmente agradável em um aplicativo Flutter, contribuindo para uma experiência de usuário mais atraente e organizada.

Vamos adicionar o card a nossa aplicação. Para isso, precisamos adicionar outra coluna dentro da nossa coluna, essa coluna será onde nossos cards ficarão.

Agora dentro da coluna adicionada, adicione uma linha.

Vamos adicionar alguns componentes.

Dentro da linha, adicione um IconButton.

Dentro da linha novamente, adicione outra coluna.

E dentro dessa coluna adicione dois textos, o primeiro texto será o nome do aluno e o segundo texto será o e-mail.

Vamos agora adicionar mais um IconButton na linha.

Vamos modificar os dois IconsButtons, os dois terão a borda e o fundo transparente, porém o ícone do primeiro será um pearson e o segundo um arron.

Ótimo, com os componentes adicionados, vamos o envolver em um card.

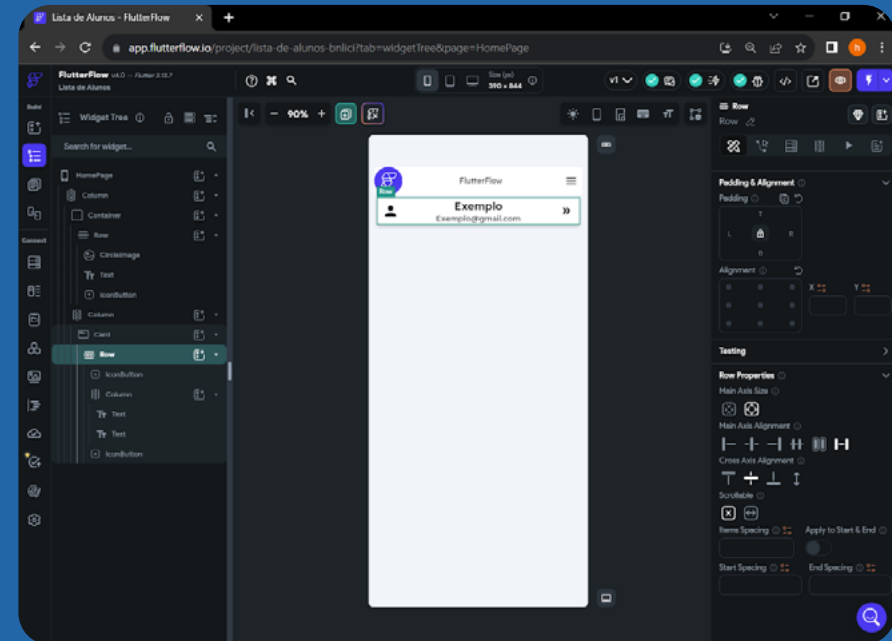
Para isso, basta clicar com o botão direito do mouse na linha, use a opção wrap widget e selecione o componente card.

Perceba que o card foi adicionado, porém o layout não está muito legal, vamos agora fazer as devidas modificações.

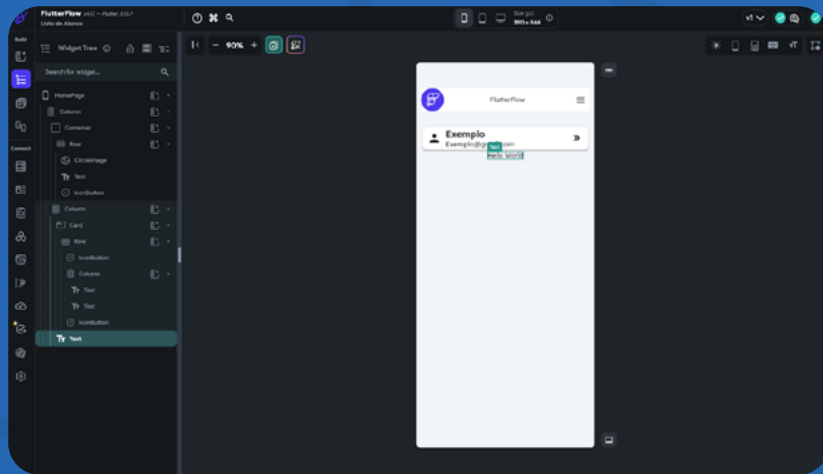
Primeiro vamos dar um destaque maior ao nome do aluno, deixando com o tamanho da fonte 20 e deixar a fonte em 600 - semi Bold.

O e-mail com o tamanho da fonte 15.

Primeiro botão com tamanho 50 e tamanho do ícone 30.

Segundo botão com tamanho 50 e tamanho do ícone 20.

Agora vamos deixar um espaçamento entre os componentes.

INVESTIGAR

Pergunte aos alunos como podemos fazer isso.

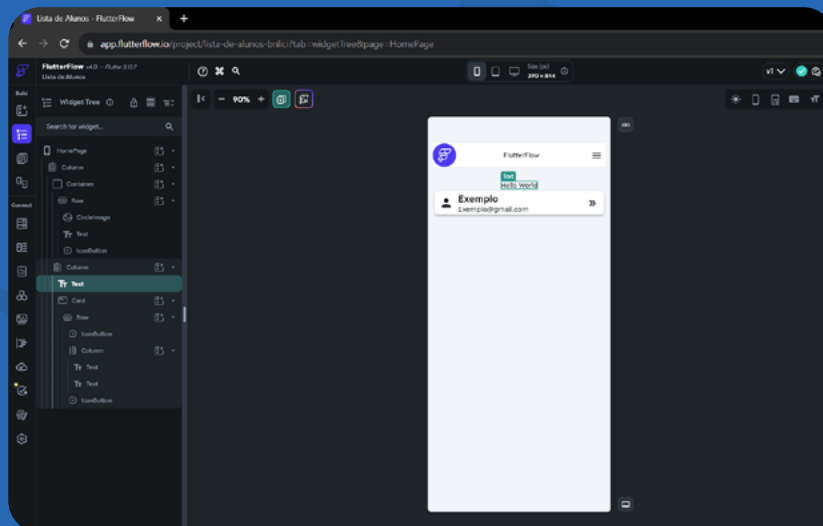
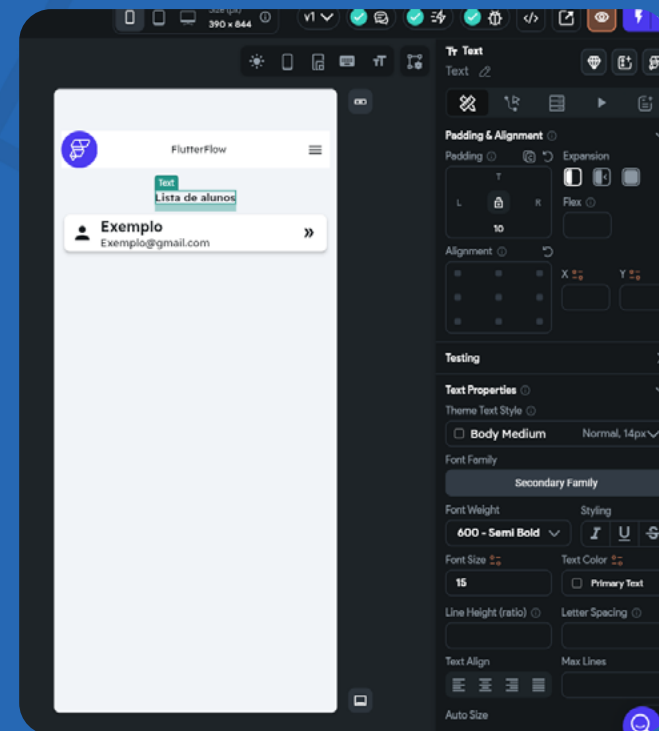
Resposta:

Usando space Between na nossa linha dentro do nosso card que envolve os componentes.

Porém os textos do meio ainda não estão legais, vamos melhorar isso.

Primeiro vamos alinhar os dois textos, usando o start nas opções de ***cross Axis Alignment.***

Altere o texto para “lista de alunos” e vamos dar um padding de 10 embaixo, tamanho da fonte de 15 e 600- semi bold.

Observe que o text ficou abaixo do card.

Pergunte aos alunos se alguém sabe como mover o texto para ficar acima do card.

Resposta: basta arrastar o texto na árvore de componentes até a coluna.

Peça para os alunos fazerem mais uma coluna abaixo do exemplo que foi criado na lista.

AGIR

AULA 04

FlutterFlow

Materiais necessários:

Notebook, acesso à internet.

Duração:

1 hora e 30 minutos

Objetivo:

Navegar nas telas.

ENGAJAR

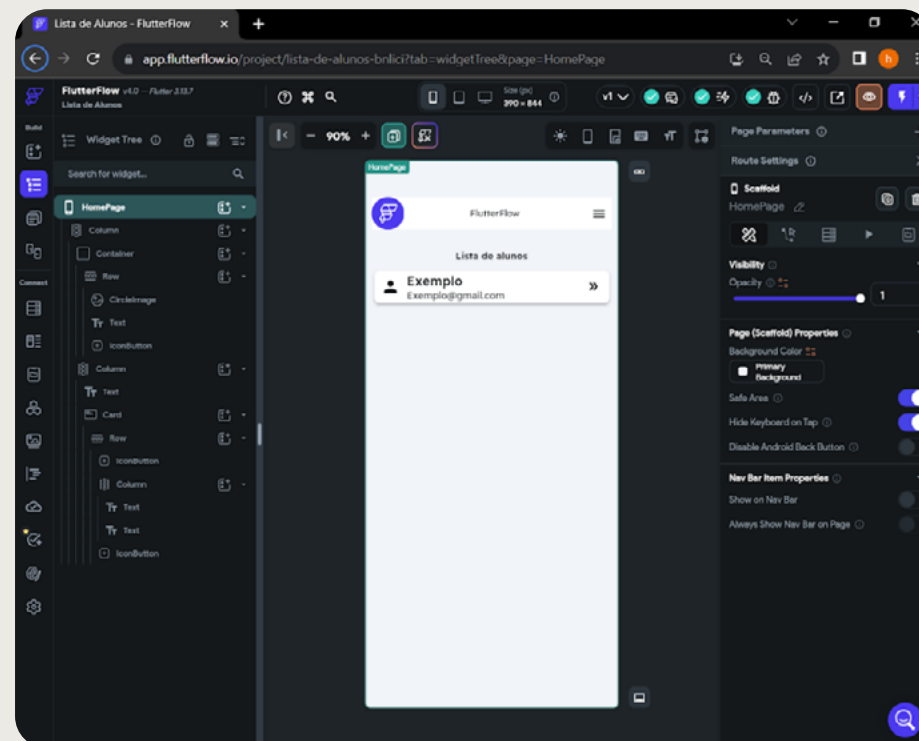
Navegação entre telas...

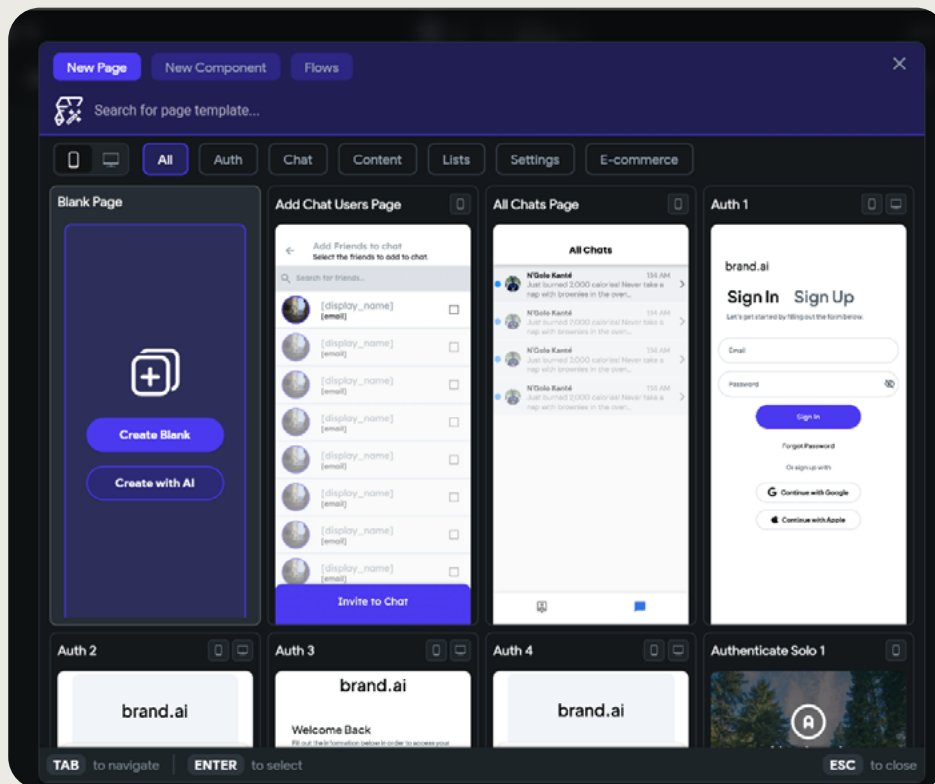
Já temos a tela inicial do nosso aplicativo, entretanto precisamos de mais telas, sendo elas:

- Login e cadastro;
- Perfil do Usuário;
- Criar um tema claro e tema escuro;
- Abrir mais informações quando clicar no card;
- Entre outras coisas.

Inicialmente vamos criar uma tela de login e dar uma função de navegação, para que essa tela direcione até a nossa tela inicial.

Vamos criar uma nova tela no nosso projeto...

Para criar uma nova tela dentro do projeto em que estamos trabalhando, basta clicar no ícone **Add page, Component, or Flow.** Aqui você pode adicionar novas páginas, componentes ou fluxo.

Na tela que apareceu podemos ver vários modelos de telas prontos, vamos criar uma tela de login e cadastro do 0.

Na primeira tela apresenta duas opções: Create Blank e Create with AI.

Create Blank

Criar uma tela do 0.

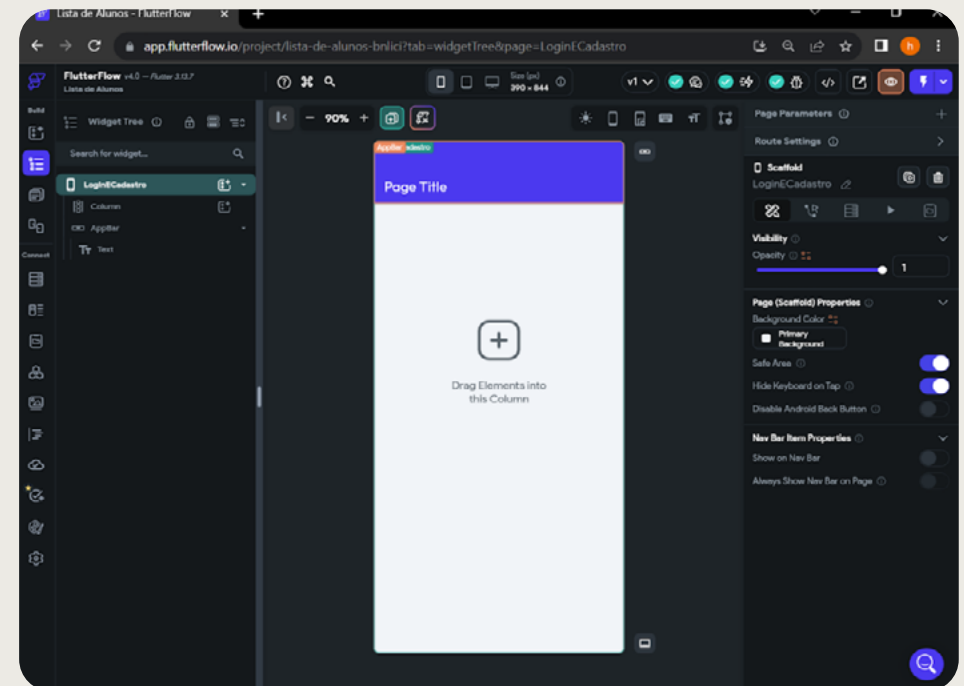
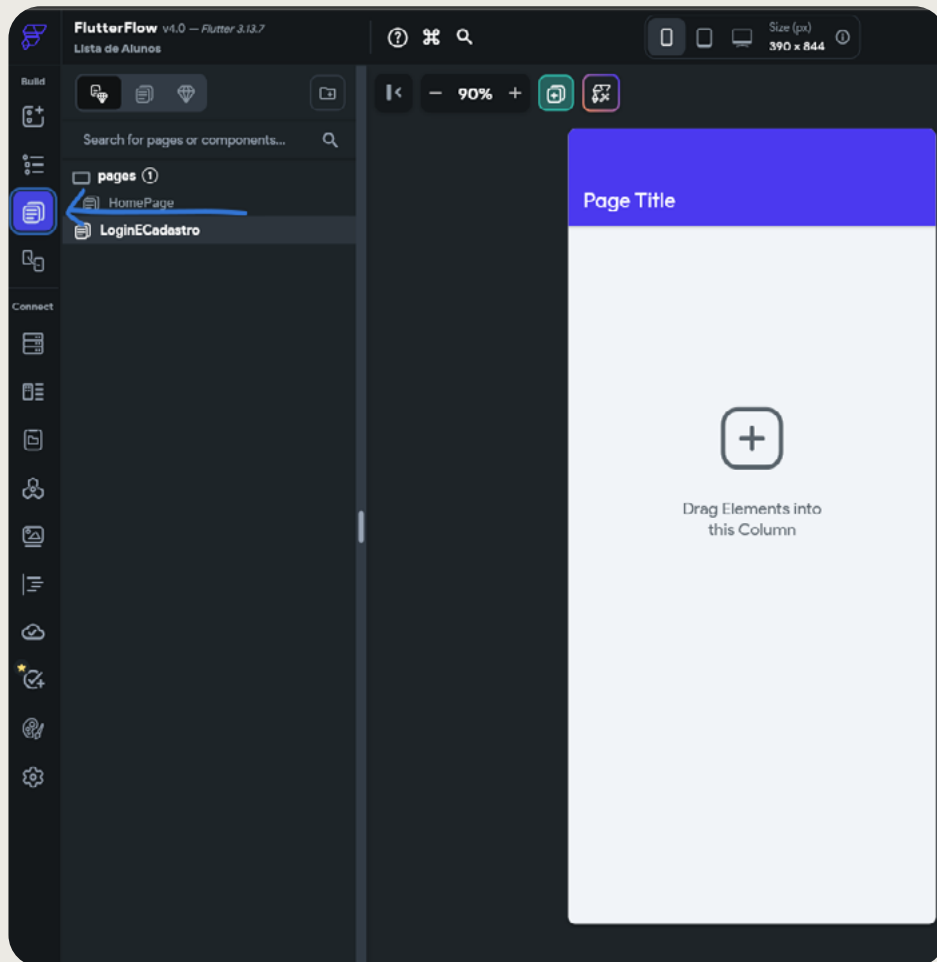
Create with AI

Criar uma tela usando uma inteligência artificial onde é possível digitar qual tema deseja criar e a IA criará um modelo para você.

Vamos criar uma tela do 0, clique em Create Blank e dê um nome a tela, a nomeie de "LoginECadastro" e clique em create page.

Nota:

FlutterFlow não aceita espaços ou acentos ao criar novas telas, instrua os alunos a deixarem o título da página sem esses caracteres.

Perceba que a nossa página inicial sumiu e agora temos uma tela nova.

Para vermos todas as telas criadas e navegar entre elas, basta clicar no botão **Page Seletor** na nossa Buil.

Veja que a nossas páginas tanto a HomePage quanto a LoginECadastro estão aqui.

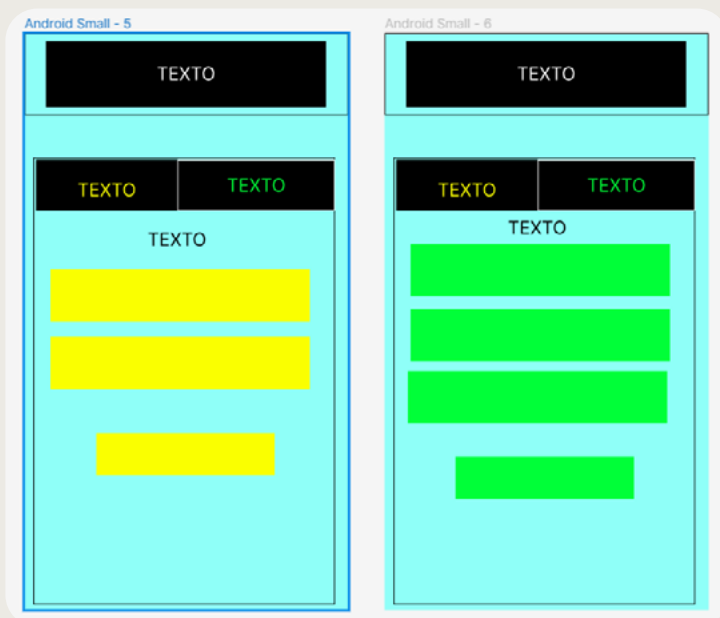
Criando a tela de login e cadastro...

A importância de ter uma tela de login e cadastro no aplicativo.

Ter uma tela de login e cadastro em aplicativos é importante por diversos motivos, tais como:

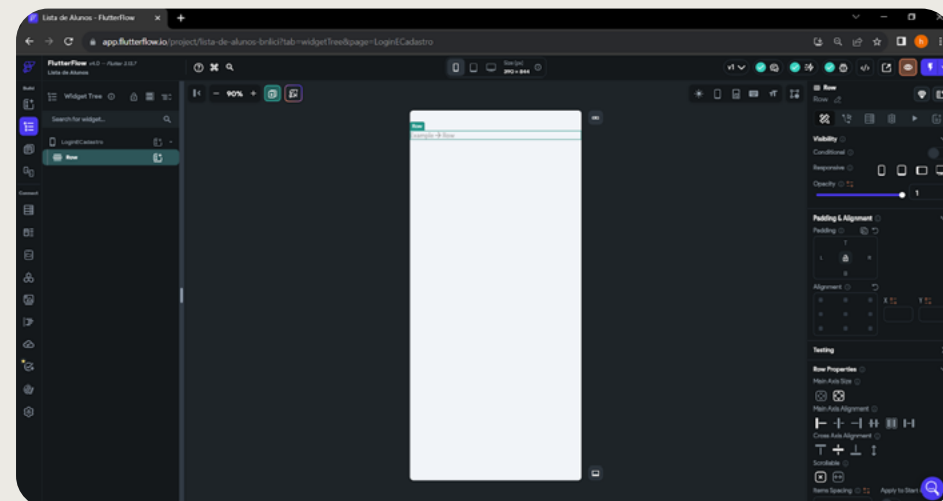
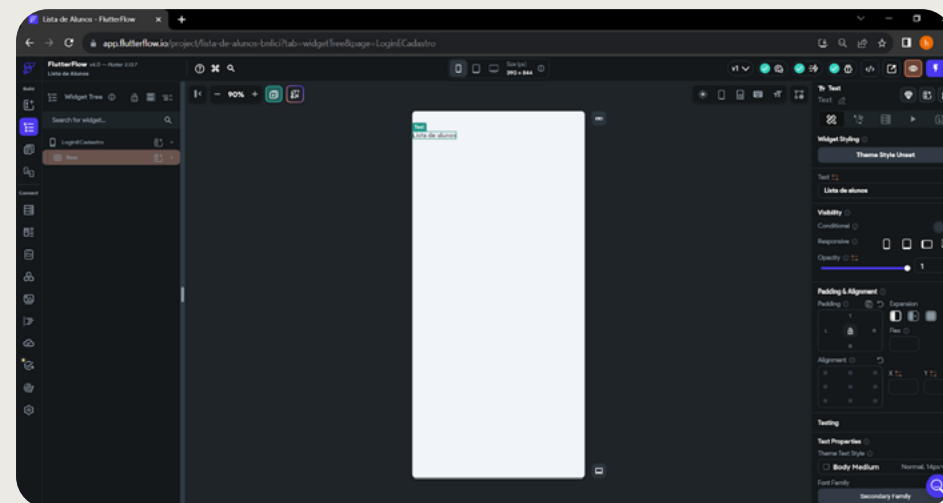
Segurança	A tela de login permite que os aplicativos autentiquem os usuários, garantindo que apenas pessoas autorizadas tenham acesso às informações e funcionalidades protegidas. Além disso, o cadastro pode ajudar na verificação da identidade do usuário.
Personalização	Ao ter um cadastro, os aplicativos podem personalizar a experiência do usuário, oferecendo conteúdos e funcionalidades de acordo com suas preferências e histórico.
Gerenciamento de dados	O cadastro permite que os aplicativos armazenem e gerenciem as informações dos usuários de forma mais eficiente, facilitando a personalização, a recuperação de dados e a comunicação com o usuário.
Comunicação e notificações	Através do cadastro, os aplicativos podem enviar notificações personalizadas e manter uma comunicação direta com os usuários, fornecendo informações relevantes e atualizações sobre o aplicativo.
Facilidade de uso	O login pode facilitar o acesso dos usuários ao aplicativo, evitando a necessidade de inserir repetidamente informações básicas, como nome de usuário e senha.

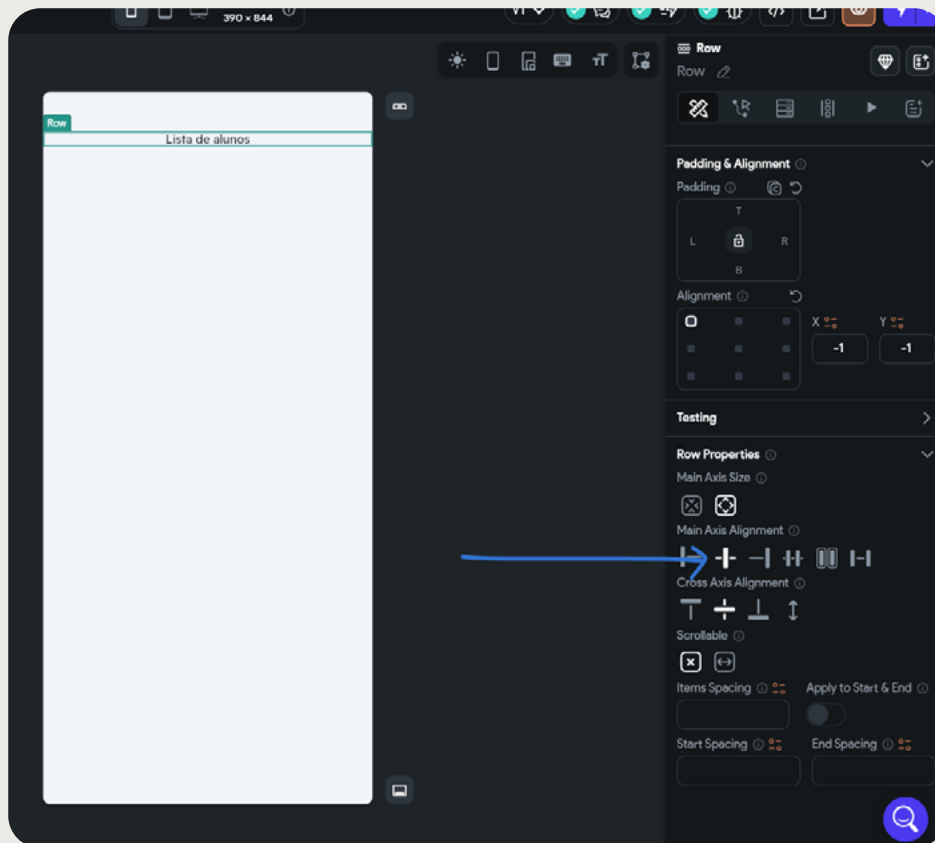
Portanto, a presença de uma tela de login e cadastro em aplicativos é essencial para garantir a segurança, personalização e eficiência na interação com os usuários.

Analisando a estrutura do layout...

Temos duas telas, sendo uma de login onde o usuário fará login caso já possua uma conta e a outra de cadastro em que o usuário criará uma nova conta.

Vamos analisar a tela login. A tela login possui um título no topo dentro de uma row, uma coluna abaixo contendo uma opção de navegação, um texto abaixo da navegação, duas caixas de texto sendo eles e-mail e senha e um botão. A tela cadastro funciona basicamente da mesma forma, porém temos três textos, sendo eles nome, e-mail e senha.

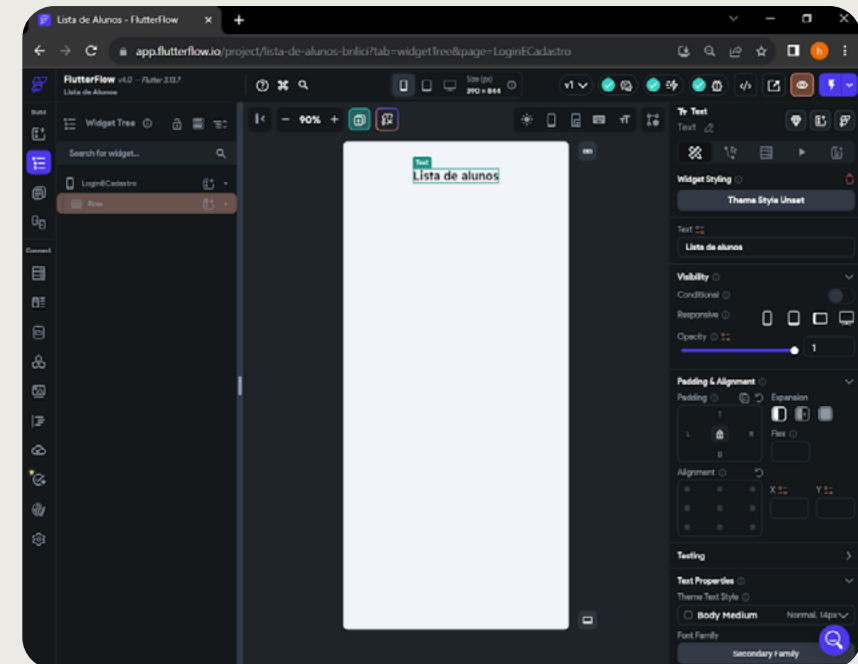
Na nossa tela **LoginECadastro,** delete os componentes que vêm padrão, assim teremos um ambiente de desenvolvimento limpo.

Vamos primeiro criar o topo da aplicação, nosso Scaffold adicione uma **row**.

Como vimos, no topo da página temos um texto, o texto será o nome do aplicativo **"Lista de alunos"**, adicione um texto dentro da linha e modifique o nome dele.

Perceba que o texto está no canto da tela e precisamos fazer algumas modificações para deixar com um melhor layout.

Vamos centralizar o texto no meio da tela, para isso basta clicar na linha que está o texto e usar a opção **center** no alinhamento.

Peça para os alunos fazerem essas modificações depois mostrem para eles.

Fonte Size: 20

Font Weight: 600 - Semi bold

Basta clicar no texto e modificar as propriedades.

Agora no nosso layout perceba que há duas opções para navegar entre o login e o cadastro, para fazer isso vamos aprender um novo componente.

TabBar...

O que é o TabBar?

O componente TabBar no FlutterFlow é uma parte fundamental para a criação de aplicativos com navegação por abas. Ele permite que os usuários alternem entre diferentes seções ou funcionalidades do aplicativo, organizando o conteúdo de forma clara e acessível.

Ao adicionar o componente TabBar em seu projeto no FlutterFlow, você pode configurar as abas desejadas e associar cada uma a um conteúdo específico. Isso facilita a navegação entre diferentes partes do aplicativo, proporcionando uma experiência intuitiva para os usuários.

Além disso, o componente TabBar oferece opções de personalização, como a escolha de ícones, rótulos e cores para cada aba, permitindo que você adapte o visual das abas de acordo com as necessidades do seu aplicativo.

Em resumo, o componente TabBar no FlutterFlow é uma ferramenta poderosa para criar uma navegação por abas eficiente e atraente em seus aplicativos Flutter.

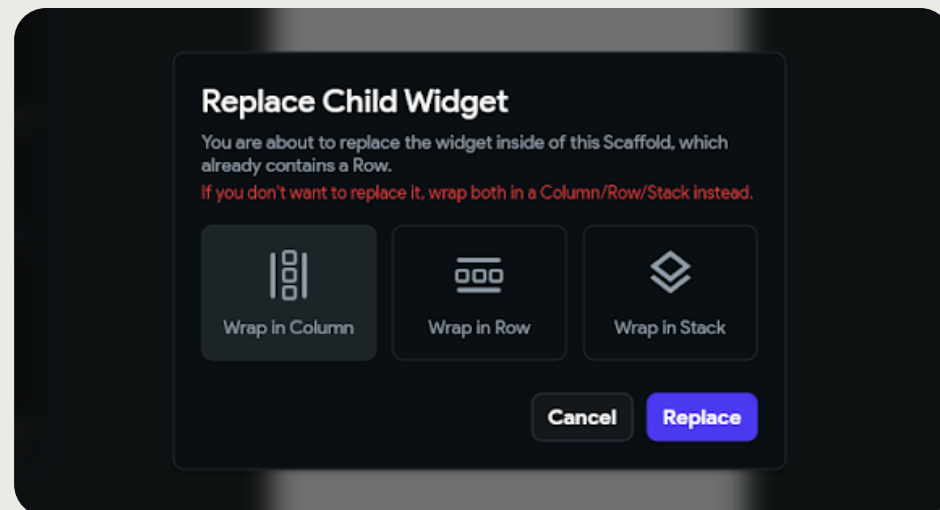
Se tentarmos adicionar a TabBar diretamente no scaffold, vai dar um erro.

Replace Child Widget

You are about to replace the widget inside of this Scaffold, which already contains a Row.

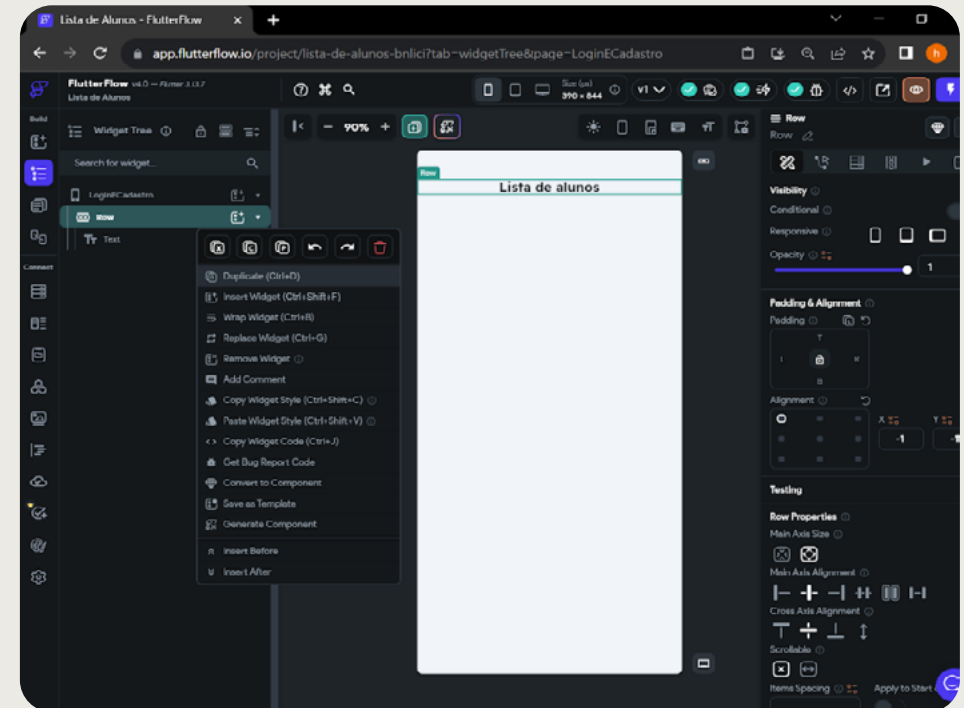
If you don't want to replace it, wrap both in a Column/Row/Stack instead.

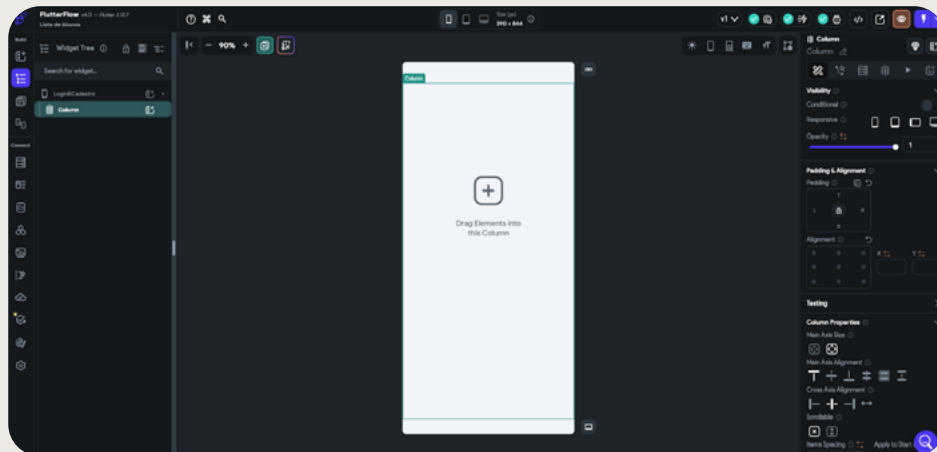
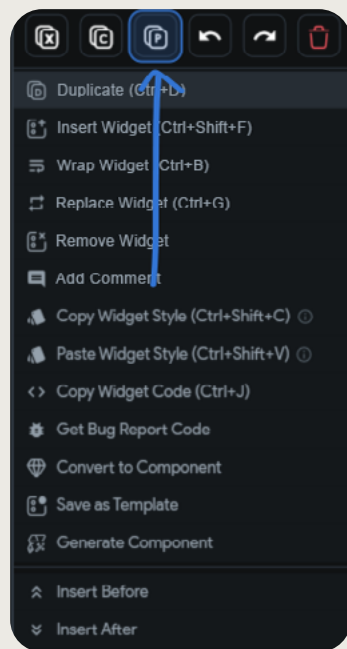
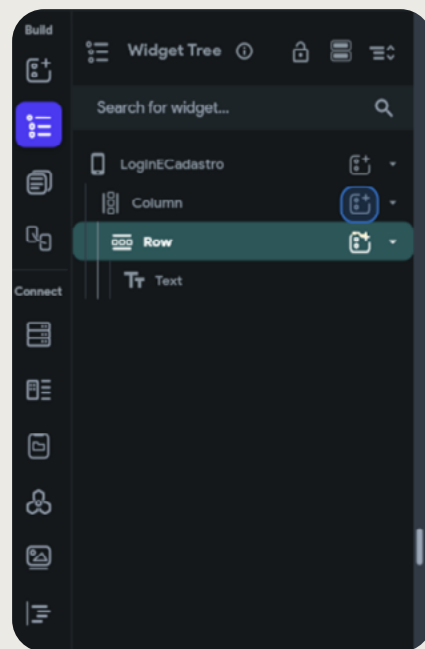
Wrap in Column | Wrap in Row | Wrap in Stack

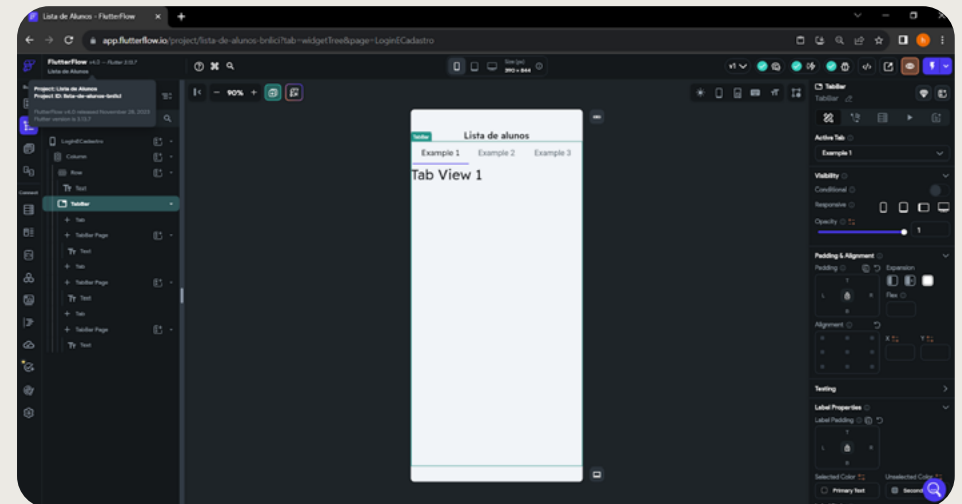
Cancel | Replace

O FlutterFlow apresenta algumas regras em que não é possível substituir um widget filho, o erro ocorre porque estamos tentando substituir o widget dentro do Scaffold que já tem uma linha.

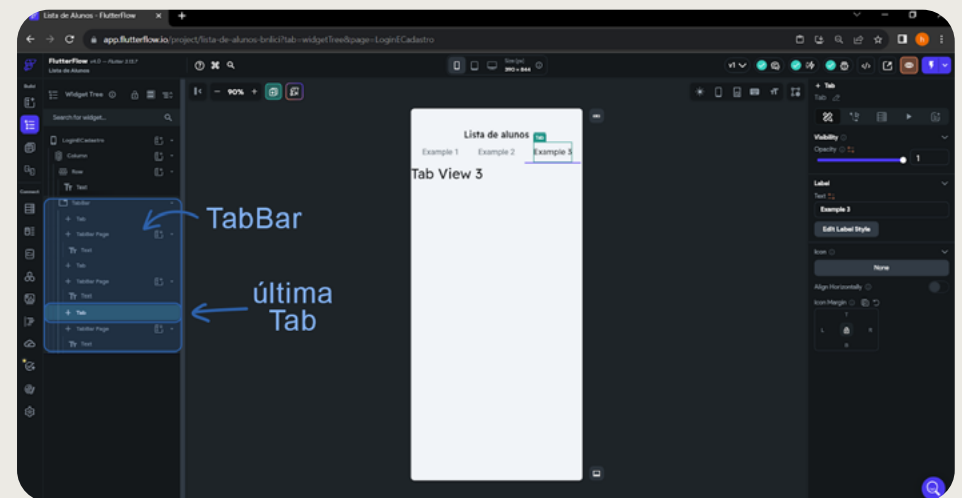
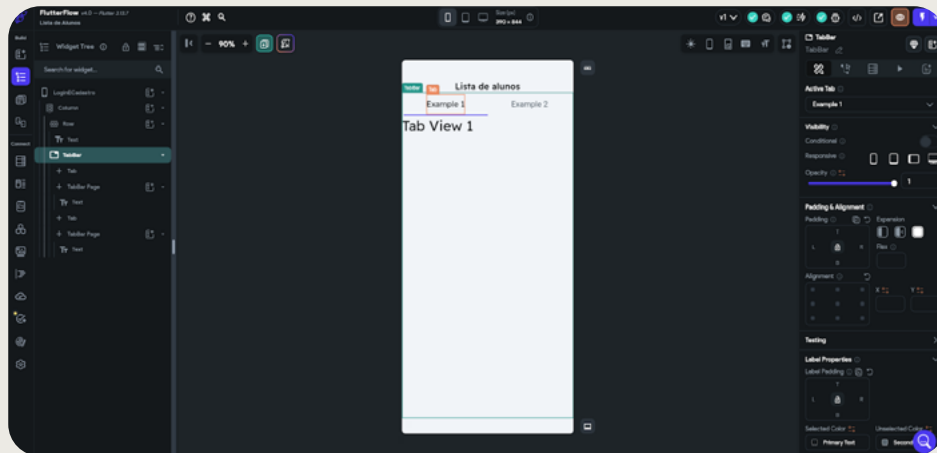
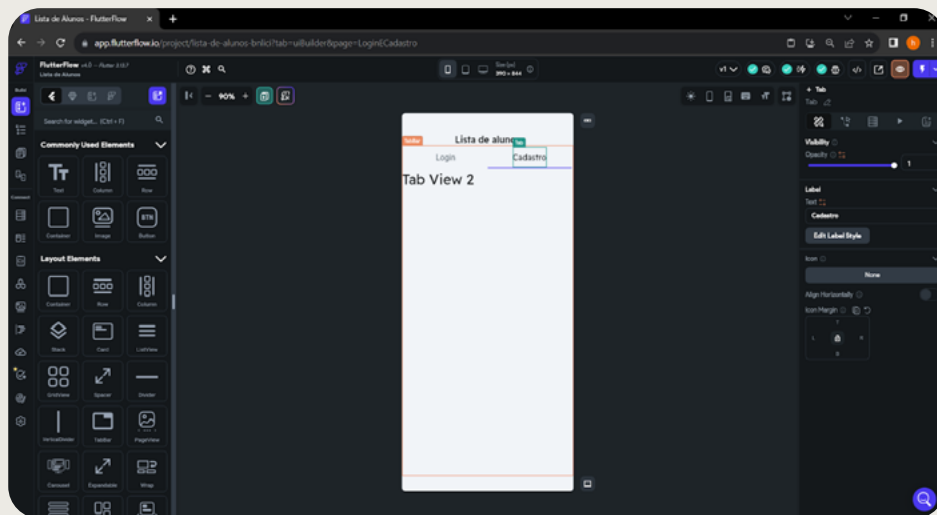
O FlutterFlow já nos dá opções para caso não queiramos substituir, para colocar ambos em uma coluna, linha ou pilha.

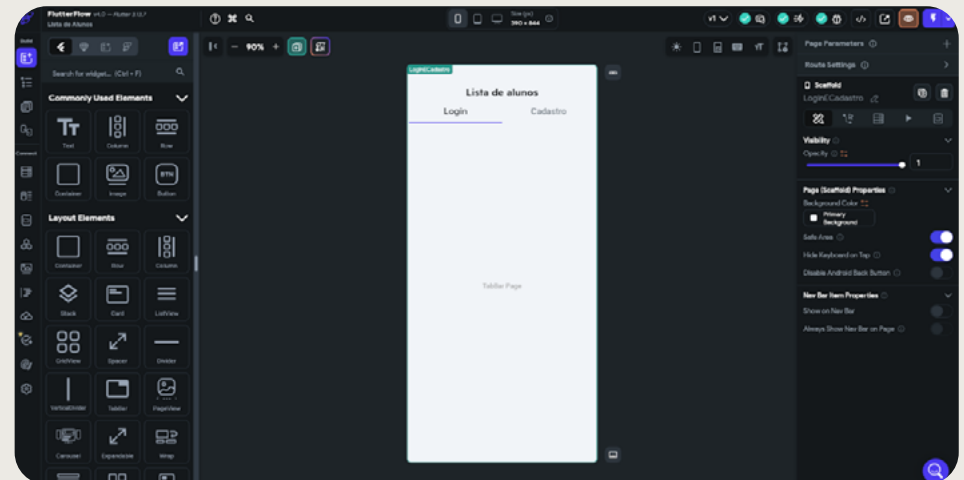
Porém vamos aprender alguns atalhos bem úteis no desenvolvimento.

Clique sobre a linha com o botão direito do mouse, aparecerão algumas opções, clique na primeira opção do topo, **Cut** (recortar).

O texto e a linha sumirão, mas calma, eles estão na área de transferência.

No nosso Scaffold adicione uma coluna.

Agora clique com o botão direito do mouse sobre a coluna e selecione a terceira opção do topo **Past** (colar).

Perceba que o texto voltou da forma que estava antes.

Vamos adicionar a nossa TabBar. Na nossa coluna, pesquise pelo componente TabBar.

O componente foi adicionado com sucesso!

Perceba que no nosso layout temos apenas duas opções, porém, na TabBar, temos três opções. Vamos excluir uma delas e alterar os títulos.

Para excluir, vamos na nossa árvore de componentes e dentro da TabBar clique com o botão direito do mouse sobre a última Tab e exclua.

Ficamos apenas com duas opções.

Para alterar o nome, vamos clicar sobre o título no scaffold e modificar o Label Text para Login e fazer o mesmo com o outro texto, porém mudar para Cadastro.

Vamos excluir o Tab View que está no meio da tela para adicionar nossos campos de textos.

Para isso basta clicar sobre ele e excluir com a tecla delete do teclado, faça isso nas duas tabs.

Vamos agora adicionar os nossos campos de textos e aprender um novo componente.

Pergunte aos alunos qual componente precisamos colocar para alinhar um campo de texto um abaixo do outro.

Resposta: Column (coluna).

Adicione uma coluna dentro da TabBar Page, faça isso nas duas TabBar Page.

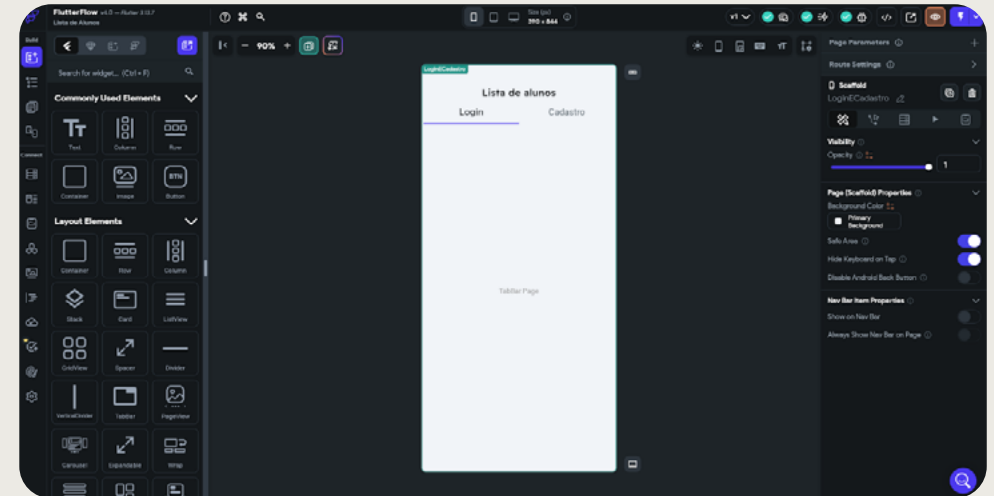
TextField...

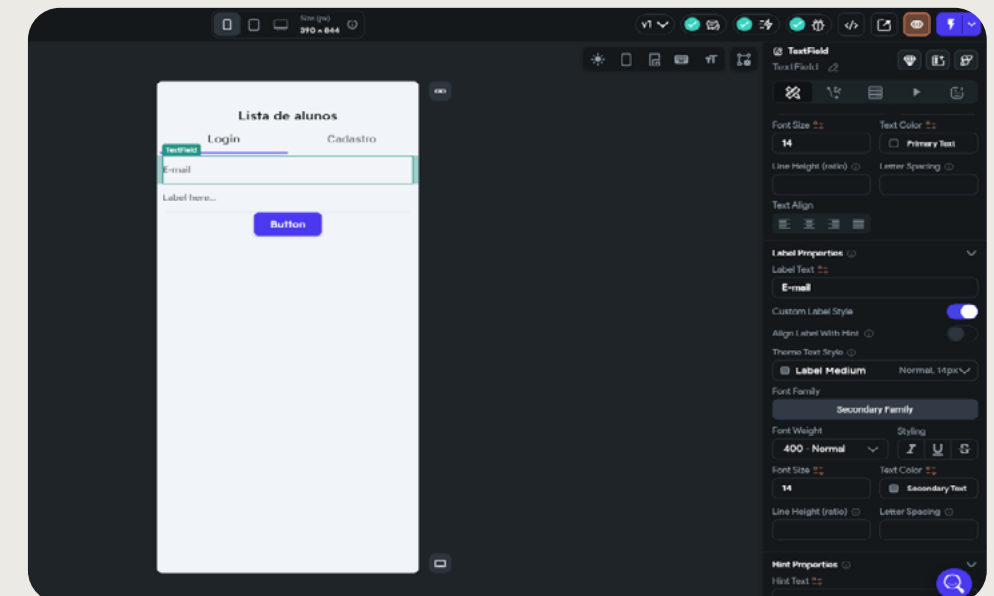
O que é?

O TextField no FlutterFlow é um widget que permite aos usuários inserirem texto em um aplicativo Flutter. Ele é usado para coletar dados de entrada do usuário, como nomes de usuário, senhas, mensagens etc. O TextField também pode ser personalizado para atender às necessidades de design específicas do aplicativo, incluindo a adição de dicas de entrada, validação de dados, formatação de texto e muito mais. Em resumo, o TextField é uma ferramenta fundamental para a interação com o usuário em aplicativos Flutter, permitindo a entrada e manipulação de texto de forma dinâmica.

Vamos adicionar dois TextFields dentro da coluna e um botão logo abaixo.

A
G
I
R

Peça para os alunos procurarem a caixinha de entrada chamada TextField, depois que encontrarem peça para colocar dentro da tela.

Vamos primeiro modificar os Label text dos TextField.

Clique no componente e procure nas propriedades por Label Text, mude para E-mail, repita o mesmo processo com o outro, porém mude para senha.

AULA 05

FlutterFlow

Materiais necessários:
Notebook, acesso à internet.

Duração:
1 hora e 30 minutos

Objetivo:
Aplicar a caixa de entrada **(textfield)**.

ENGAJAR

Conhecendo o TextField...

No FlutterFlow você pode editar várias propriedades de um TextField, para personalizá-lo de acordo com as necessidades do seu aplicativo. Aqui estão algumas das coisas que você pode editar em um TextField no FlutterFlow:

Placeholder	O texto de espaço reservado (placeholder) é exibido dentro do campo de texto quando ele está vazio. Você pode editar o placeholder para fornecer uma dica sobre o tipo de informação que deve ser inserida no campo.
Estilo de texto	Você pode personalizar o estilo do texto inserido no TextField, incluindo a cor, o tamanho da fonte, o estilo da fonte (negrito, itálico etc.), o alinhamento e muito mais.
Validação	Você pode adicionar validação ao TextField para garantir que os dados inseridos pelos usuários atendam a determinados critérios, como um formato específico de e-mail, um número mínimo de caracteres, entre outros.
Ações	É possível definir ações que ocorrem quando o usuário interage com o TextField, como submeter o formulário, limpar o campo, entre outros.
Teclado	Você pode personalizar o tipo de teclado que é exibido quando o usuário interage com o TextField, como um teclado numérico, de e-mail, de senha, entre outros.

Essas são apenas algumas das propriedades que podem ser editadas em um TextField no FlutterFlow. A plataforma oferece uma série de opções de personalização para que você possa adaptar o TextField de acordo com as necessidades e o estilo do seu aplicativo.

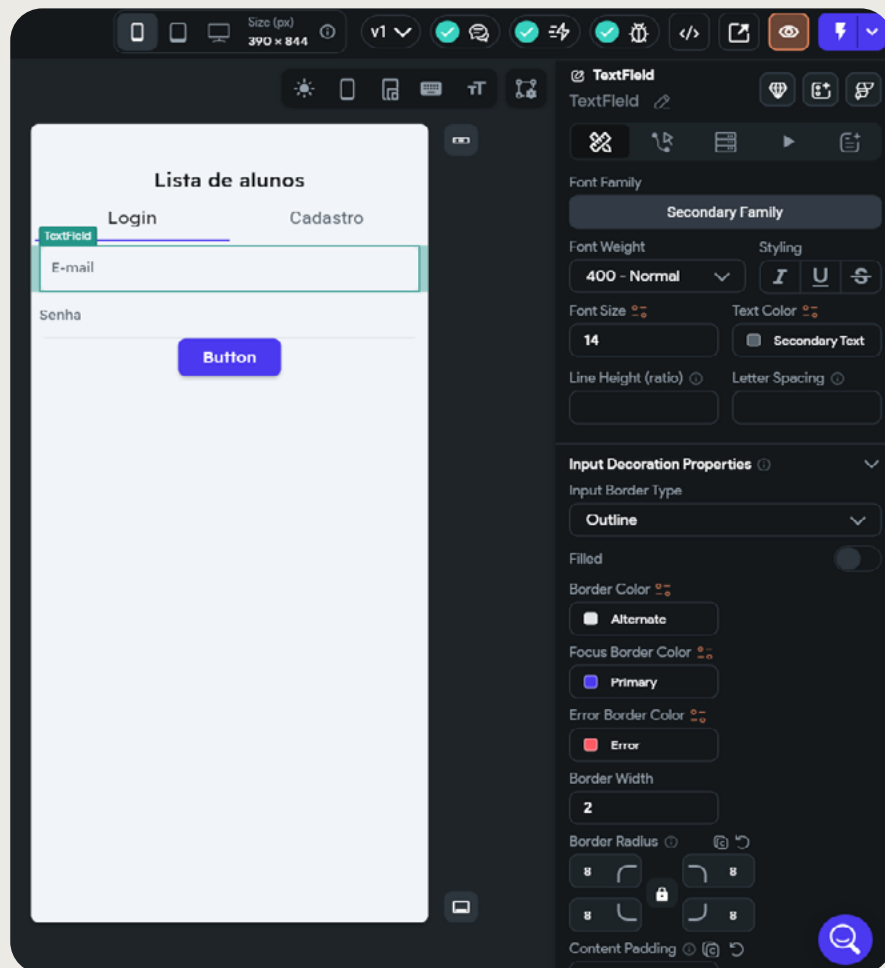
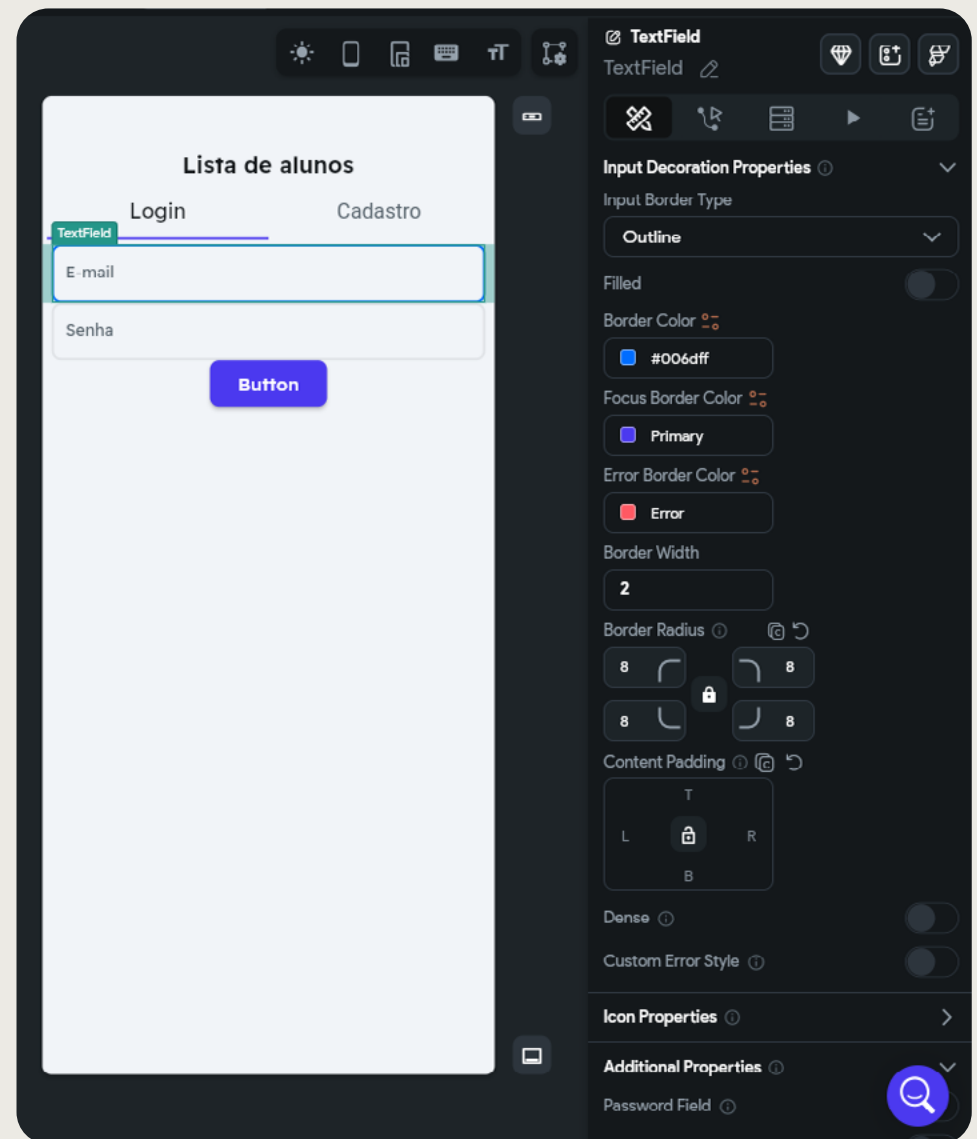
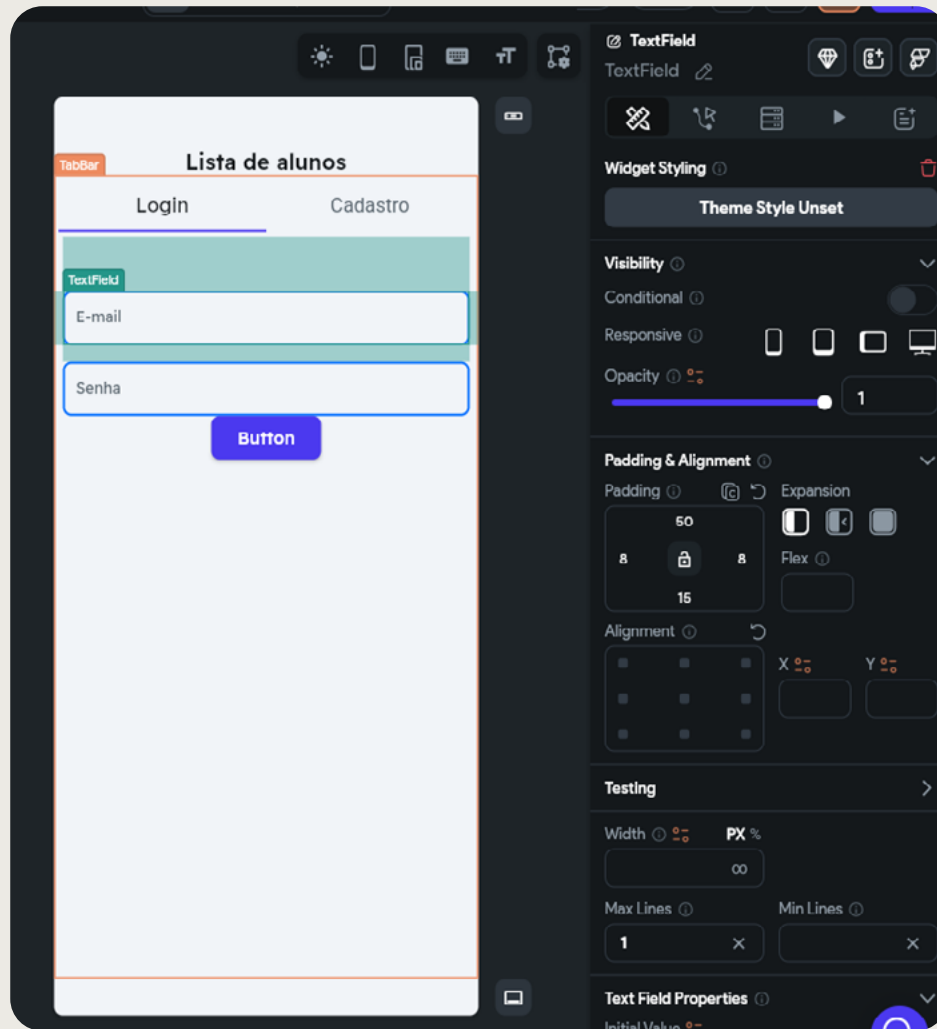
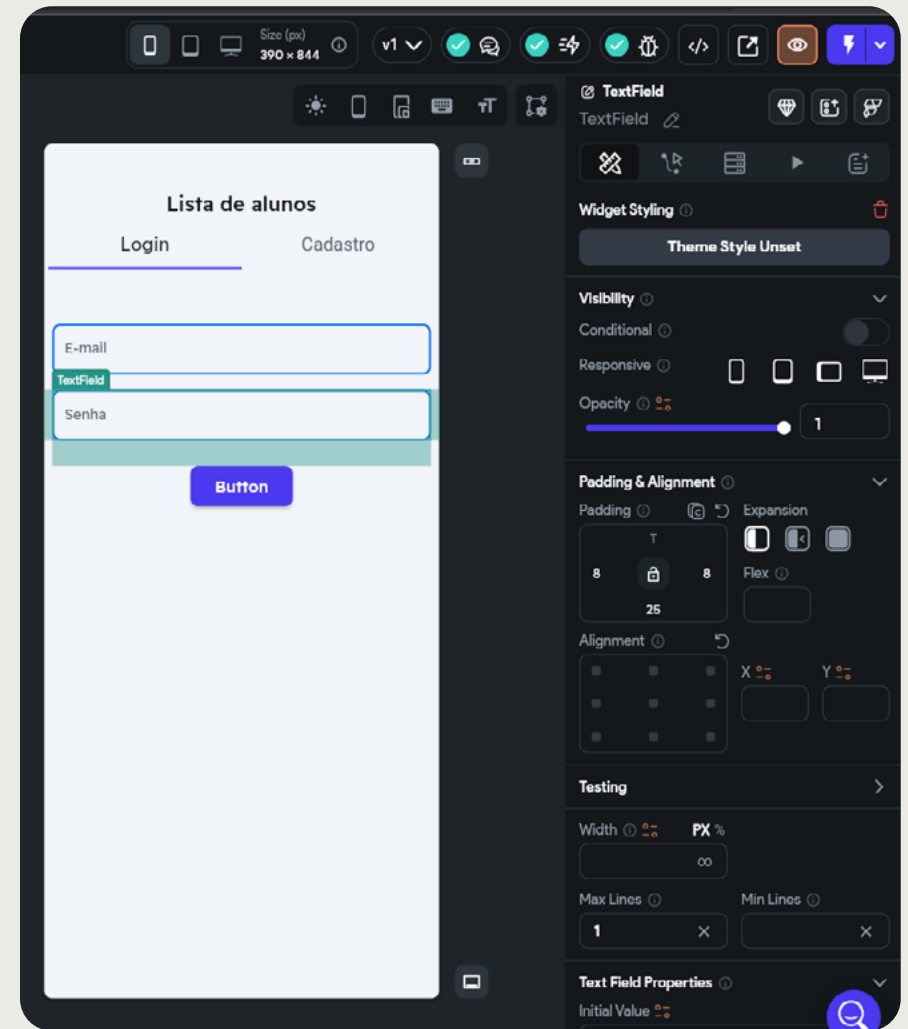
LaberlText...

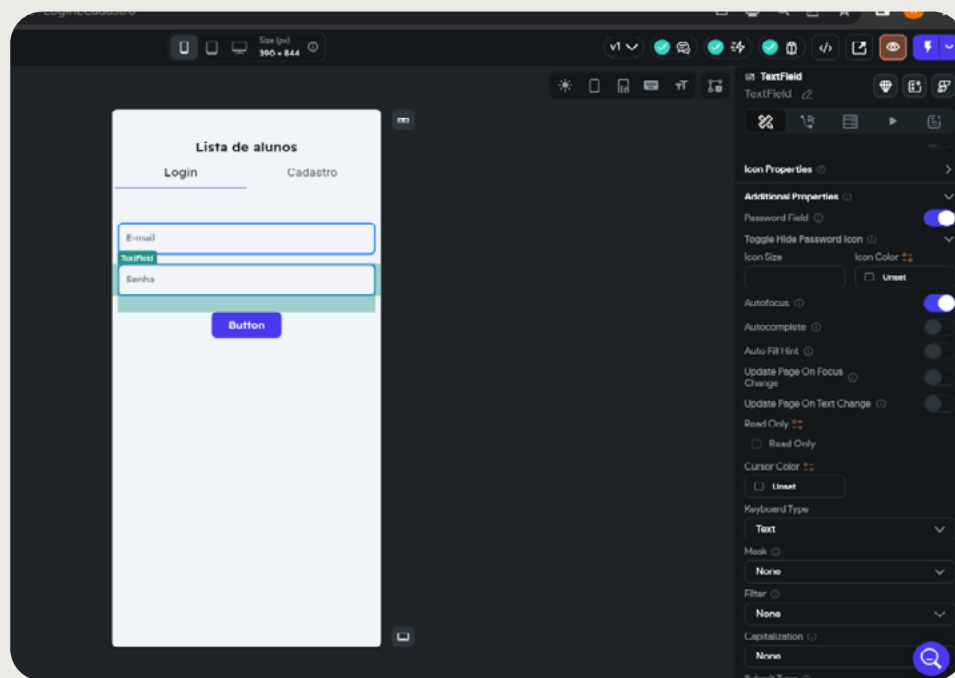
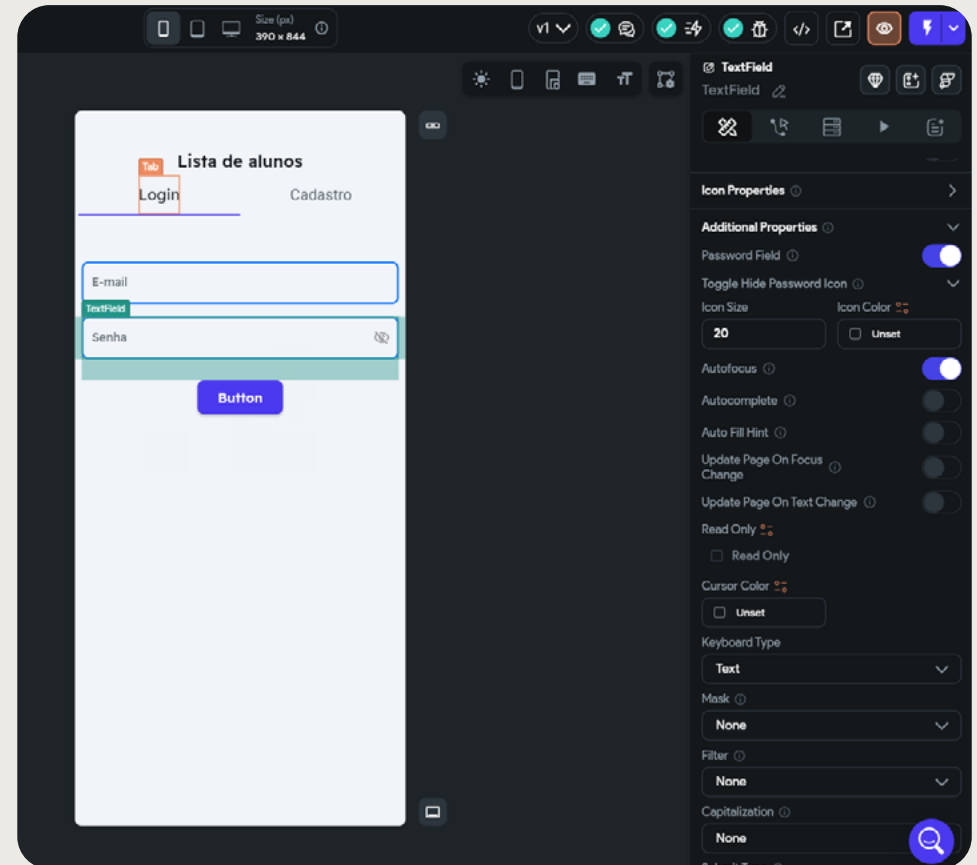
No FlutterFlow, o labelText em um TextField é o texto que aparece como uma dica para o usuário sobre o tipo de informação que deve ser inserida no campo de texto. Ao editar o LabelText, você pode personalizar o texto para orientar os usuários sobre o que é esperado que eles insiram no campo de texto.

Aqui estão algumas coisas que você pode editar no LabelText no FlutterFlow:

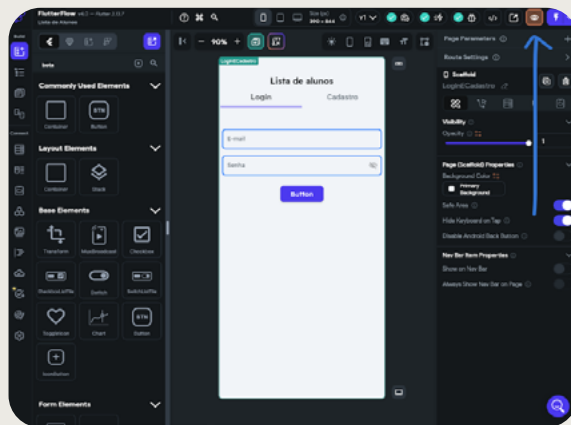
Texto	Você pode alterar o texto do LabelText para fornecer uma descrição clara e concisa sobre o que deve ser inserido no campo de texto, por exemplo, "Nome", "E-mail", "Senha" etc.
Estilo	Você pode personalizar o estilo do texto do LabelText, como a cor, o tamanho da fonte, o estilo da fonte (negrito, itálico etc.) e o alinhamento.
Tradução	Se o seu aplicativo for multilíngue, você pode editar o LabelText para fornecer traduções em diferentes idiomas, permitindo uma experiência personalizada para os usuários de diferentes regiões.

Ao editar o LabelText no FlutterFlow, você pode criar uma experiência de usuário mais intuitiva e amigável, orientando os usuários sobre o que é esperado deles ao interagir com o campo de texto.

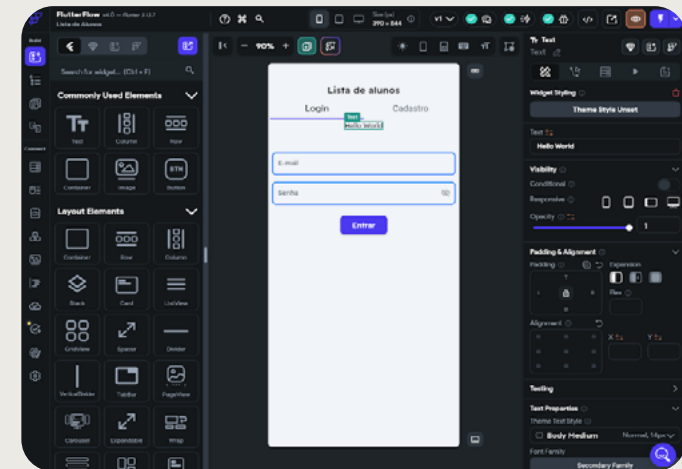
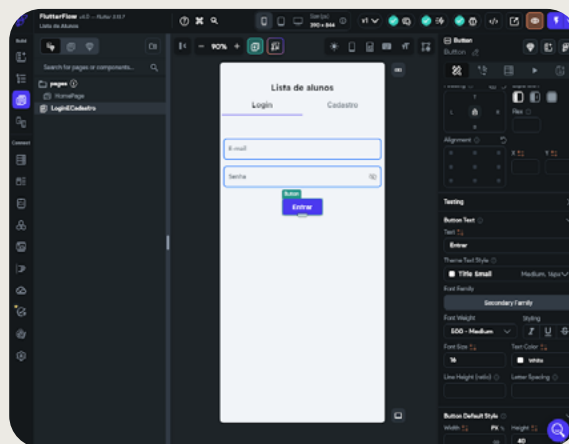
Vamos fazer um layout simples para o nosso login, vamos modificar o estilo da borda nas propriedades do TextField. Procure por **Input Decoration Properties** e altere o **Input Border Type** para **Outline**, faça isso nos dois campos de textos.

Agora vamos alterar a cor das bordas para um azul nos dois campos de texto.

Veja que os dois campos de textos estão muito próximos um do outro, deixe da seguinte forma no primeiro padding do campo e-mail.

Vamos dar um espaço entre o botão e o campo senha:

Agora vamos adicionar uma propriedade adicional, vamos definir campo senha como de fato um campo de senha e assim adicionar um ícone para ocultar a senha.

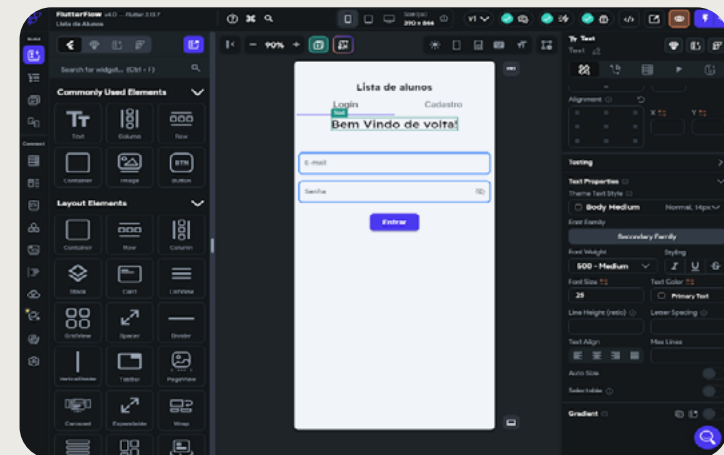
Clique no campo senha e procure por Additional Properties e marque a opção Password Field.

O nosso ícone já foi adicionado, porém ele está sem um tamanho definido. Defina um tamanho de 20 no campo Icon Size e perceba que o ícone acabou de aparecer.

INVESTIGAR

Vamos testar e ver se está funcionando.

Para ver um preview do aplicativo basta clicar em um ícone de olho presente no topo do ambiente de desenvolvimento e um emulador será aberto.

O nosso campo de senha está funcionando corretamente.

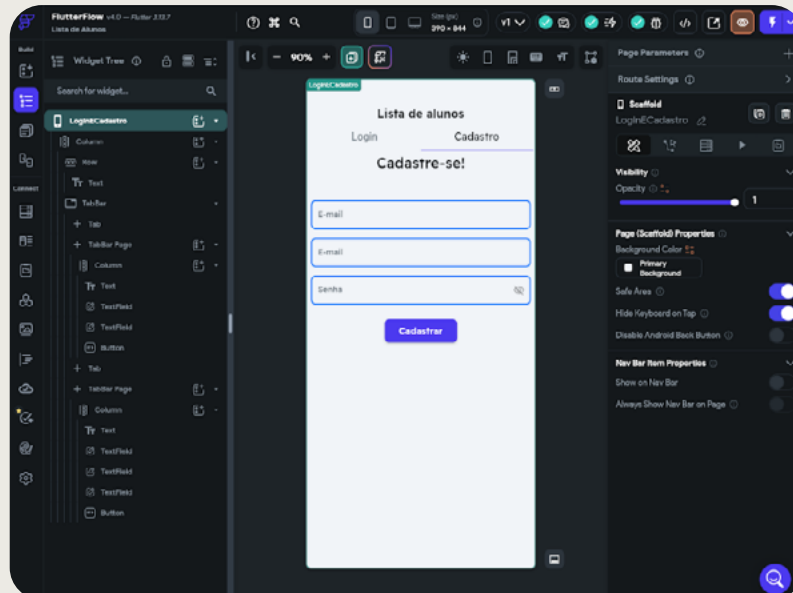
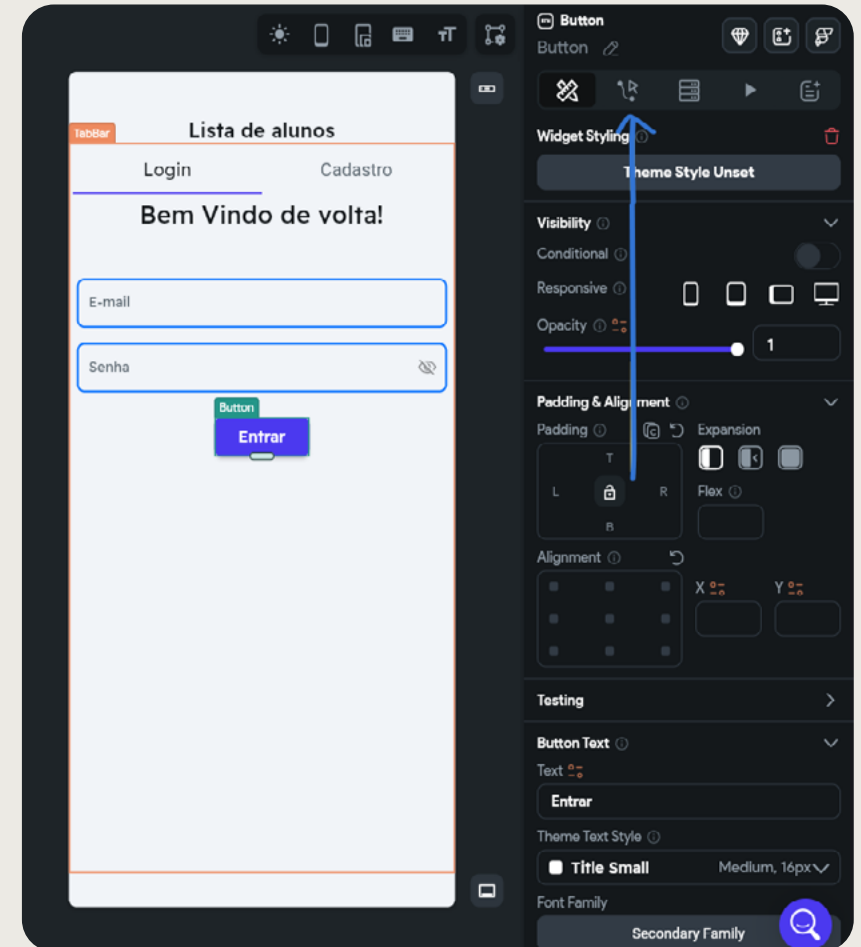
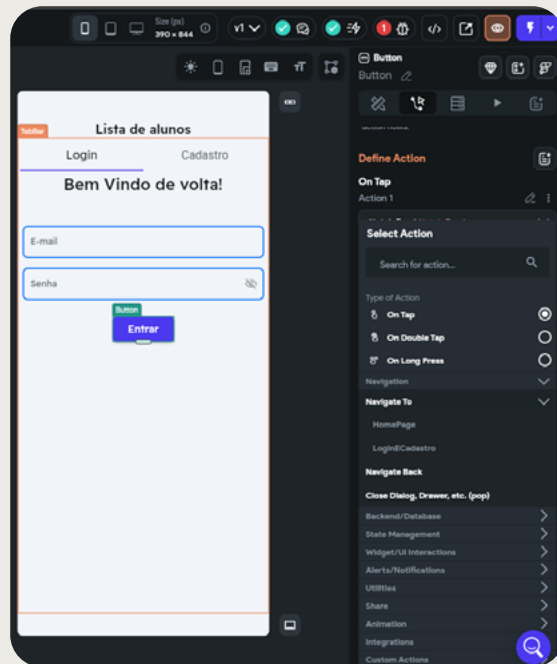
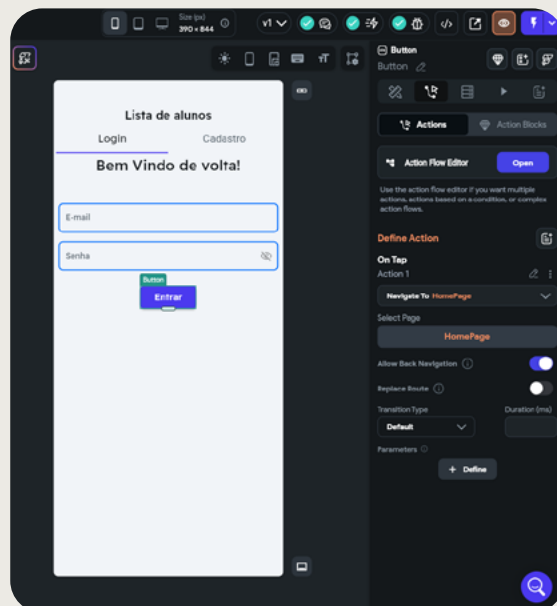
Vamos modificar o texto que do botão para “Entrar” basta clicar no botão e editar a propriedade Text.

Agora vamos adicionar um texto “bem-vindo de volta” acima do campo e-mail e para fazer isso basta arrastar o componente Text para o lugar desejado.

Vamos dar um tamanho para a fonte de 25 e um font weight de 500-medium.

A G I R

Hora de fazer o cadastro. Peça para os alunos copiarem o login e colar no cadastro.

Vá para a árvore de componentes e copie toda a coluna clicando com o botão direito do mouse e escolhendo a opção Copy, basta clicar uma vez e será copiado.

Agora vamos para a TabBar da segunda página, excluir a coluna que está lá e colar a coluna que copiamos como a opção paste dentro da TabBar, perceba que todo o conteúdo foi colado. Vamos editar e adicionar mais um campo de texto. Faça as seguintes modificações:

- “Boas-vindas” para “Cadastre-se”;
- Botão “entrar” para “Cadastrar”;
- Duplicar o campo “E- mail” modificando o mesmo para “Nome” (basta clicar com o botão direito e escolher a opção Duplicate;
- Excluir a altura do padding.

Com as telas criadas, vamos criar a função para navegar até a nossa HomePage. Para fazer isso, basta clicar no botão e adicionar uma ação para ele, uma ação que será executada ao clicar no mesmo.

Clique em Add Action e selecione Navigate To.

Selecione a página que quer navegar, no caso é a HomePage.

Está pronta a nossa ação e navegação.

Teste com os alunos usando o Preview App.

AGIR

Teste com os alunos usando o Preview App.

Caso dê tempo, peça para eles fazerem alguma tela de aplicativo.

AULA 06

FlutterFlow

Materiais necessários:

Notebook, acesso à internet.

Duração:

1 hora e 30 minutos

Objetivo:

Colocar ideias de aplicativos em prática.

ENGAJAR

Firebase...

O Firebase é uma plataforma de desenvolvimento de aplicativos fornecida pelo Google. Ele oferece uma ampla gama de serviços e recursos para ajudar no desenvolvimento, medição, melhoria e crescimento de aplicativos móveis. No contexto do FlutterFlow, o Firebase é usado como um backend para armazenar dados, autenticação de usuários, hospedagem de arquivos e muito mais.

Aqui estão algumas informações sobre o Firebase no FlutterFlow:

Configuração do Firebase

Para adicionar um novo projeto do Firebase a um projeto existente do FlutterFlow, você pode acessar as configurações e integrações do projeto no FlutterFlow e selecionar a opção de configuração do Firebase. Em seguida, siga as etapas fornecidas para configurar o Firebase corretamente.

Regras do Firebase no FlutterFlow

As regras do Firebase no FlutterFlow são usadas para controlar o acesso aos dados armazenados no Firebase. No entanto, algumas pessoas relataram que as regras do Firebase no FlutterFlow não funcionam corretamente. Certifique-se de revisar e testar suas regras para garantir que elas estejam configuradas corretamente.

Documentação e recursos

O FlutterFlow oferece documentação e recursos detalhados sobre como usar o Firebase em seus projetos. Você pode encontrar tutoriais, vídeos e exemplos que o ajudarão a entender e utilizar os recursos do Firebase no FlutterFlow.

Vantagens do firebase

O Firebase oferece uma série de vantagens para o desenvolvimento de aplicativos, como armazenamento em nuvem escalável, autenticação de usuários, análise de dados, notificações push, hospedagem de arquivos e muito mais. Ele fornece uma solução abrangente para as necessidades de backend do seu aplicativo.

Integração com o FlutterFlow

O FlutterFlow facilita a integração do Firebase em seus projetos. Com algumas etapas simples de configuração, você pode conectar seu projeto do FlutterFlow ao Firebase e começar a aproveitar os recursos e serviços oferecidos pelo Firebase.

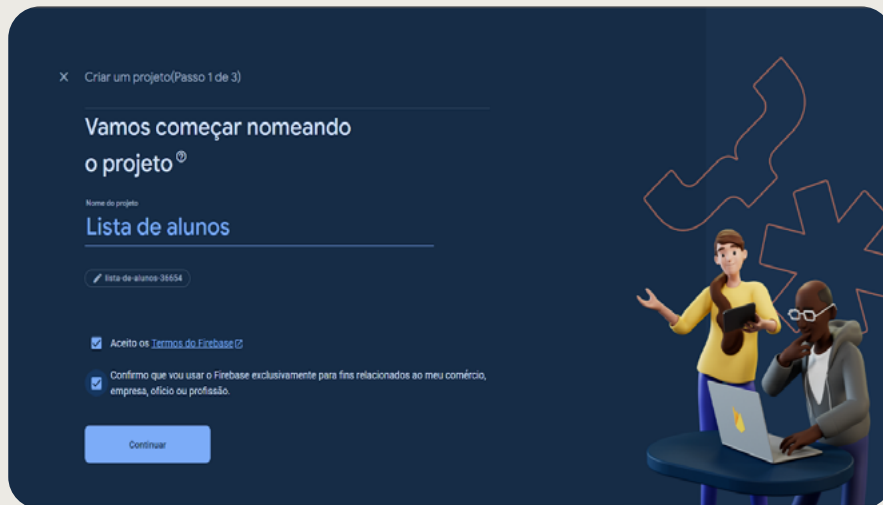
Vamos usar o firebase para fazer a autenticação do nosso login e cadastro. Para isso, precisamos primeiro possuir uma conta no Firebase.

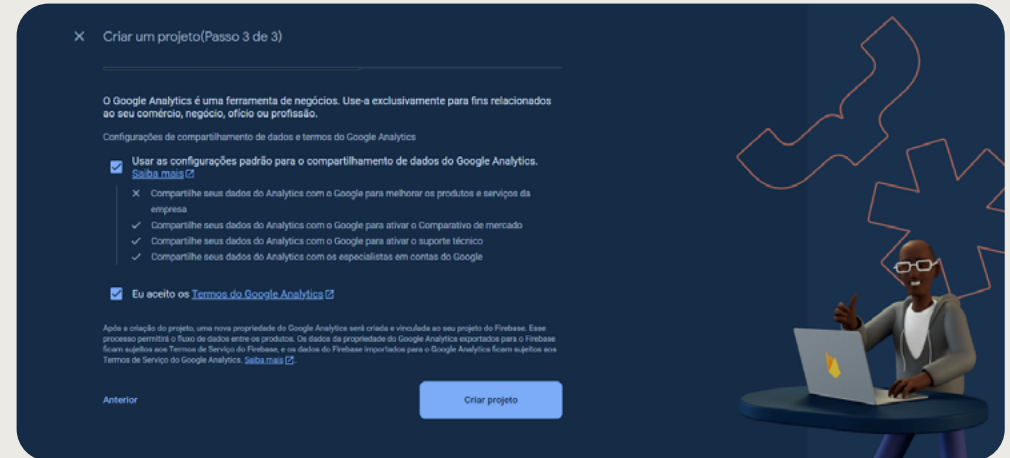
Auxilie os alunos para criar uma conta.

Acesse o site oficial do firebase (**firebase.google.com**).

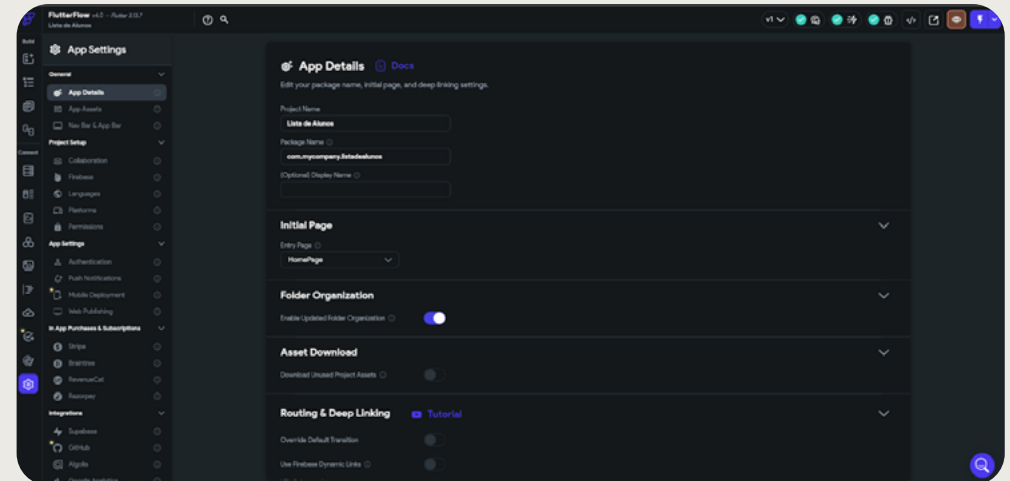
Vamos criar o nosso projeto.

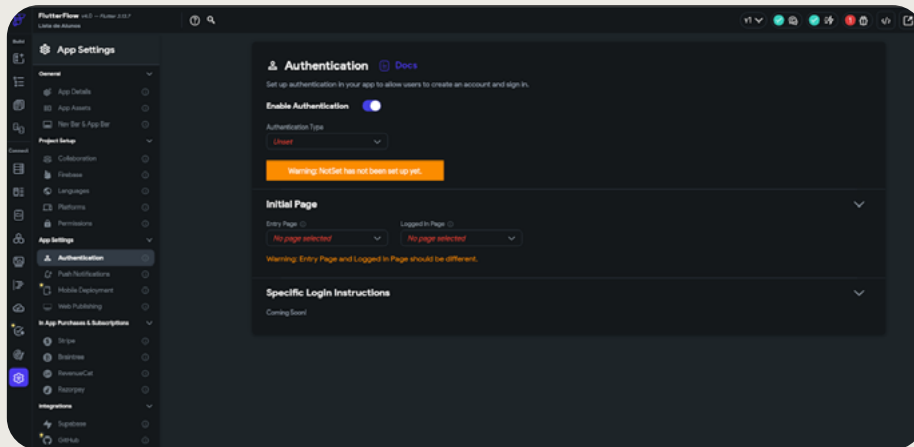
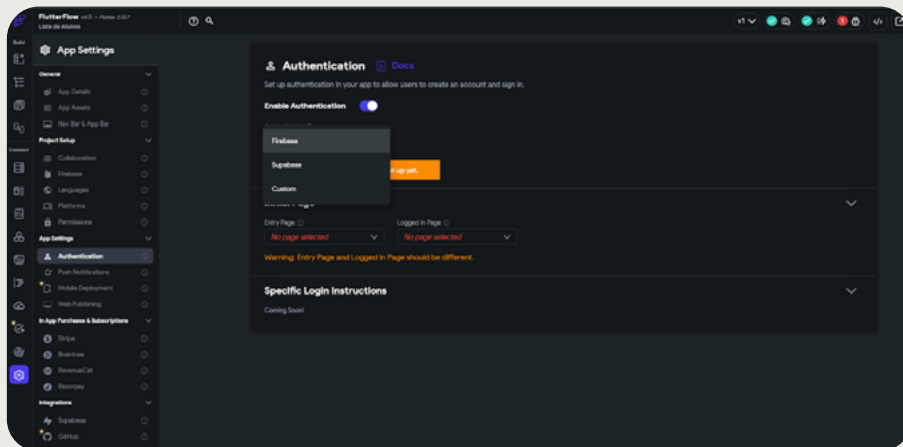
Clique em (Criar um projeto).

Dê um nome ao projeto e aceite os termos.

Continue...

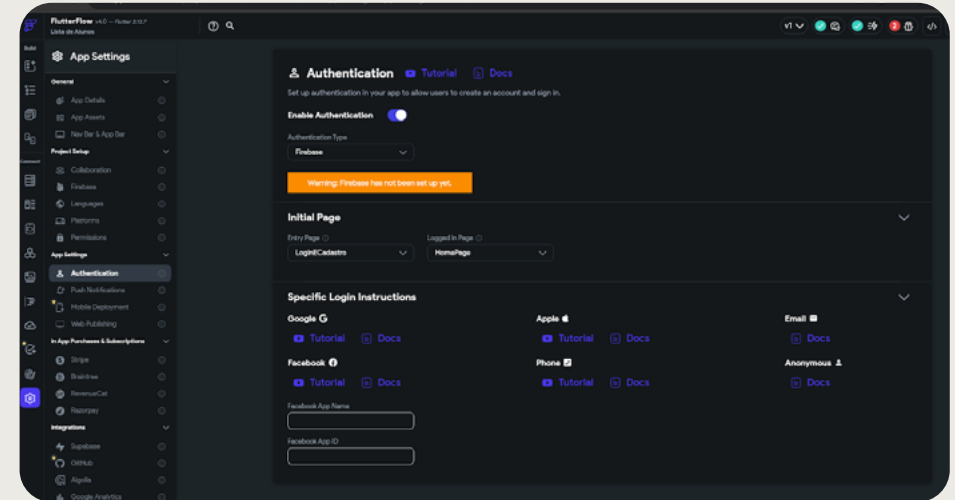
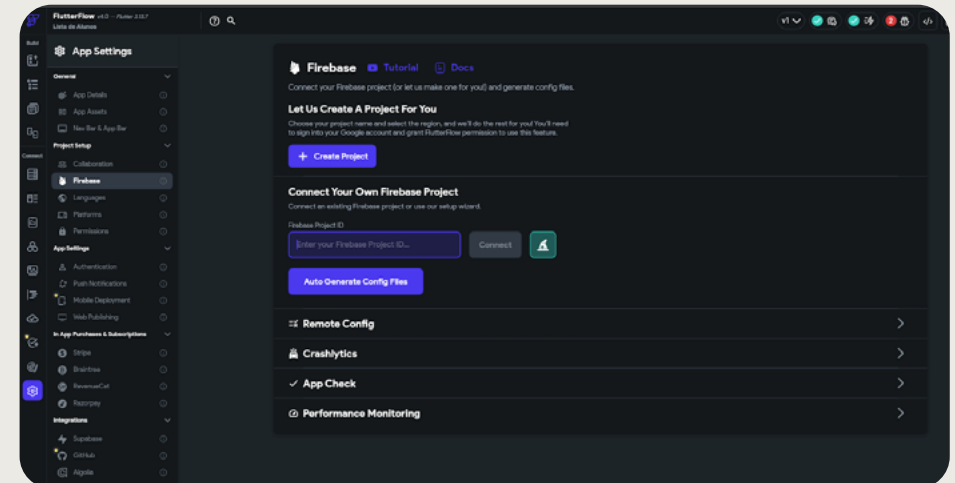
Novamente aceite os termos e crie o projeto.

Quando finalizar, pronto, o projeto foi criado com sucesso e já podemos usufruir das funcionalidades do firebase.

Vamos ao FlutterFlow para fazer a ligação, nosso objetivo é fazer a autenticação do login.

Clique nos símbolos de engrenagem no nosso **Build**.

Podemos ver diversas funcionalidades que podemos nos aprofundar futuramente, porém no momento estamos focando na autenticação.

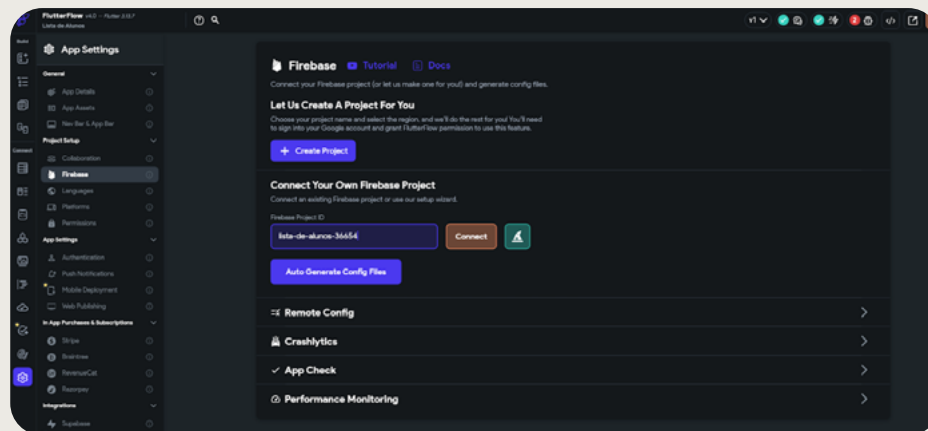
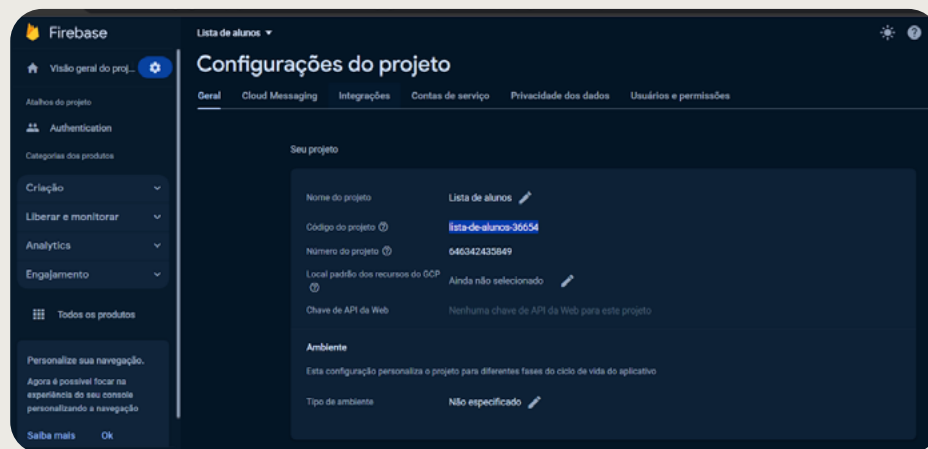
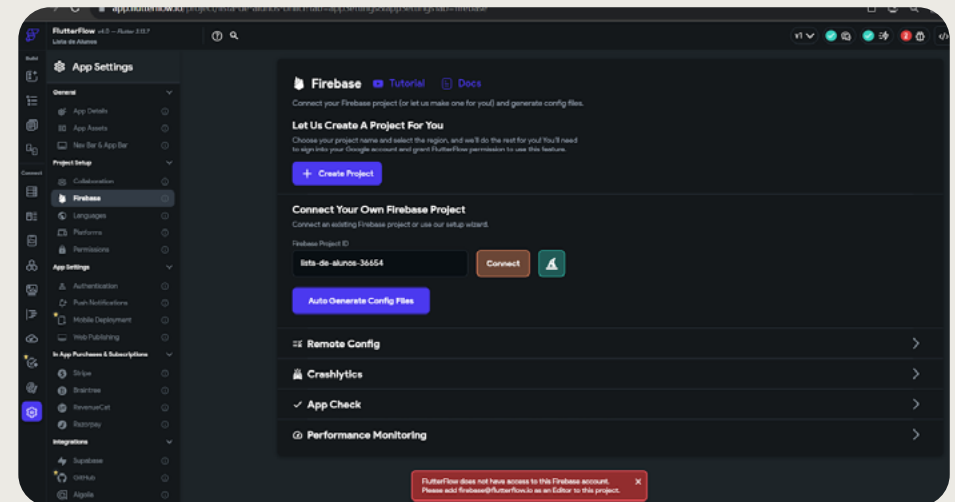
Vá em App Settings - Authentication e habilite a autenticação.

Devemos agora configurar todo o ambiente na primeira opção “Authentication Type”. Vamos falar por onde estamos fazendo a autenticação, marque a opção firebase.

Initial Page:

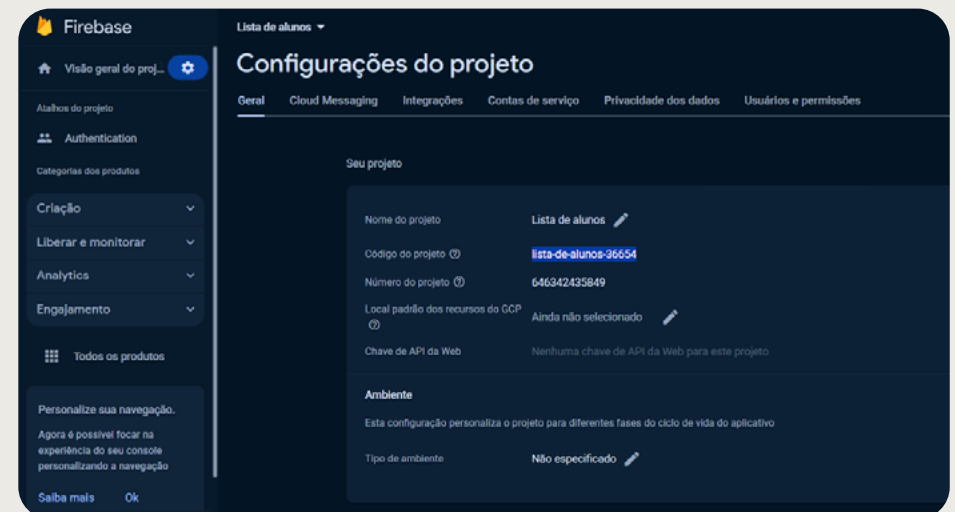
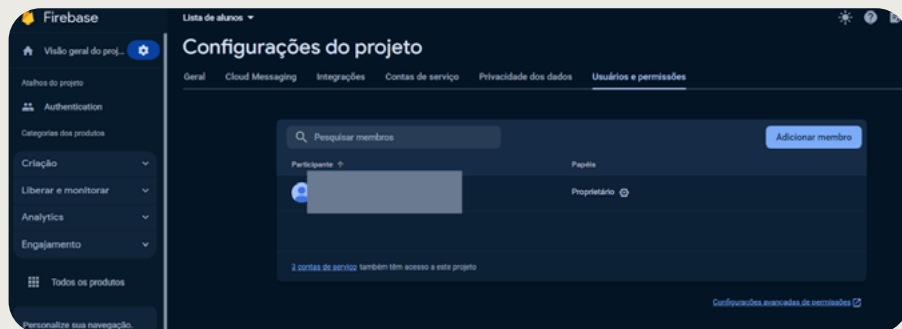
Vamos falar qual é a nossa página de entrada e qual é a nossa página logada.

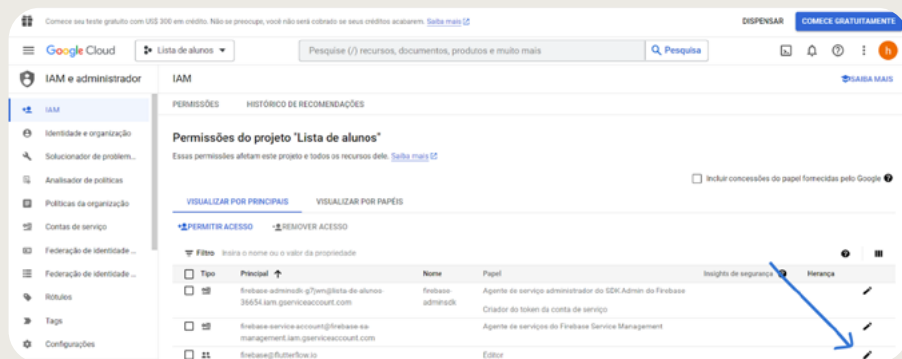
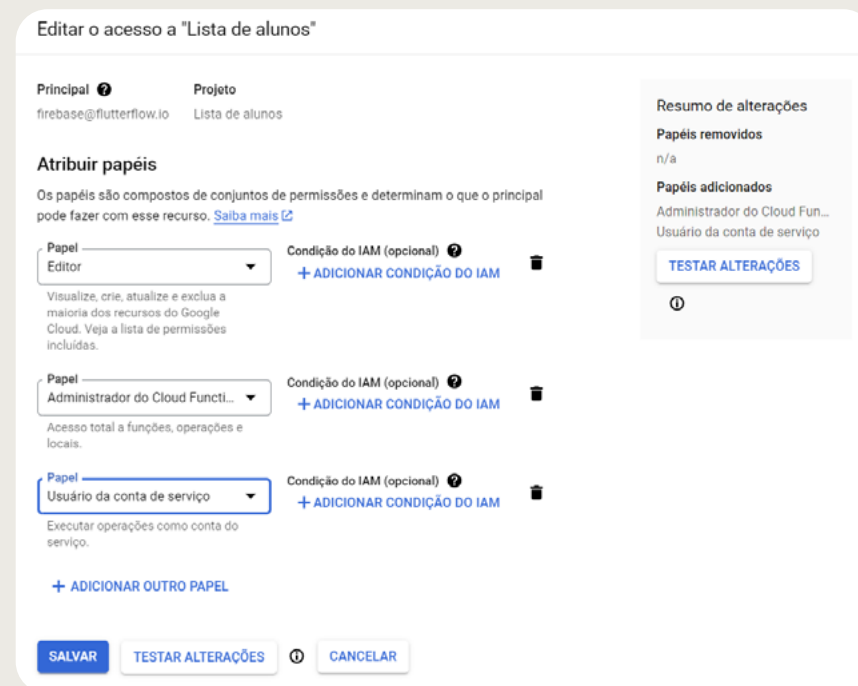
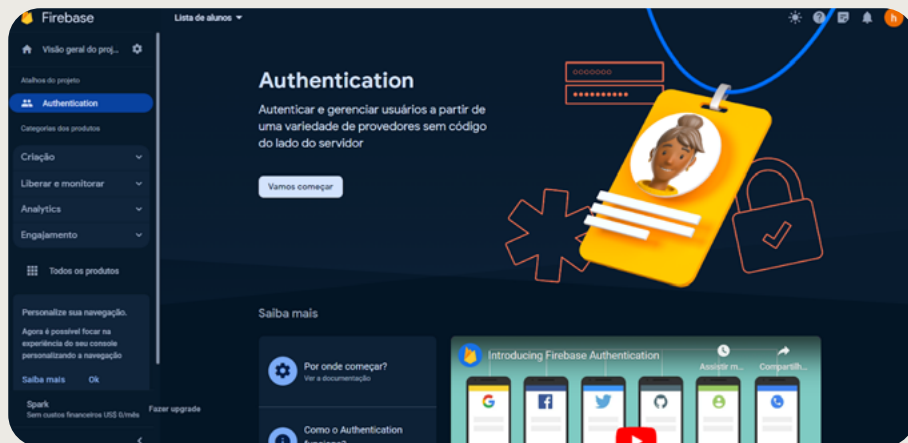
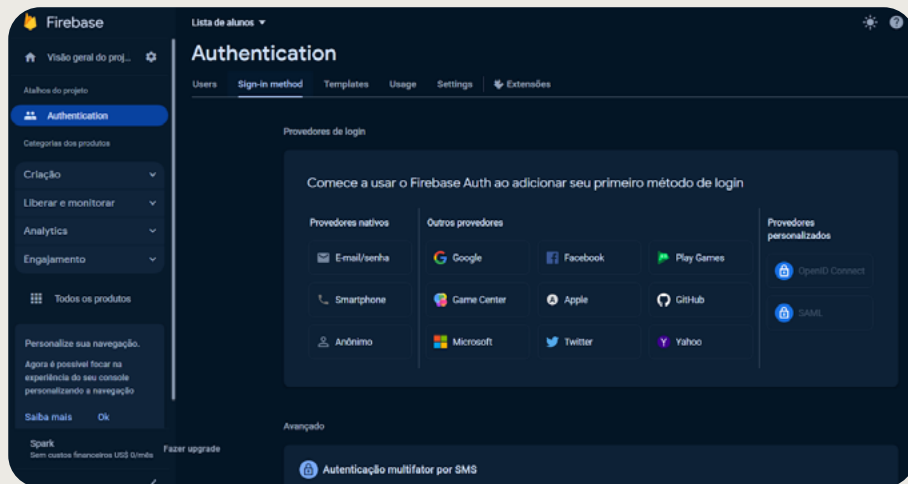
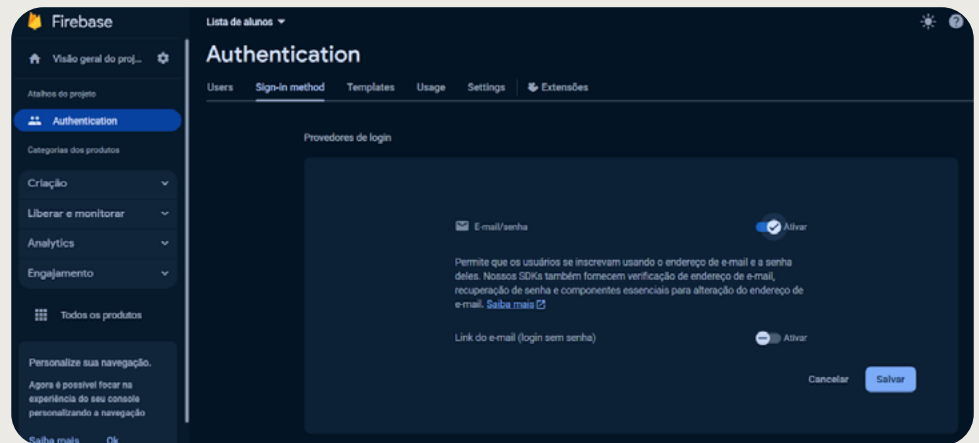
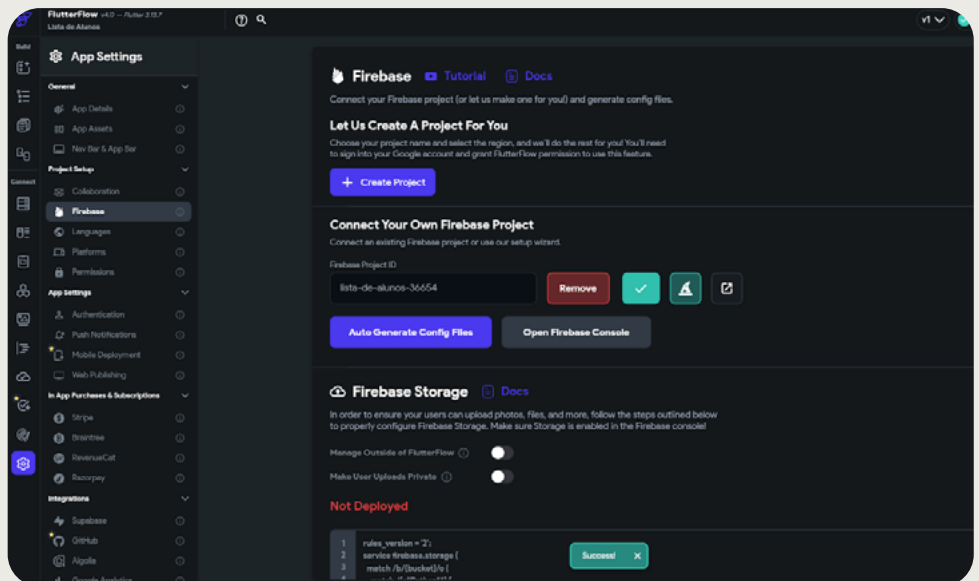
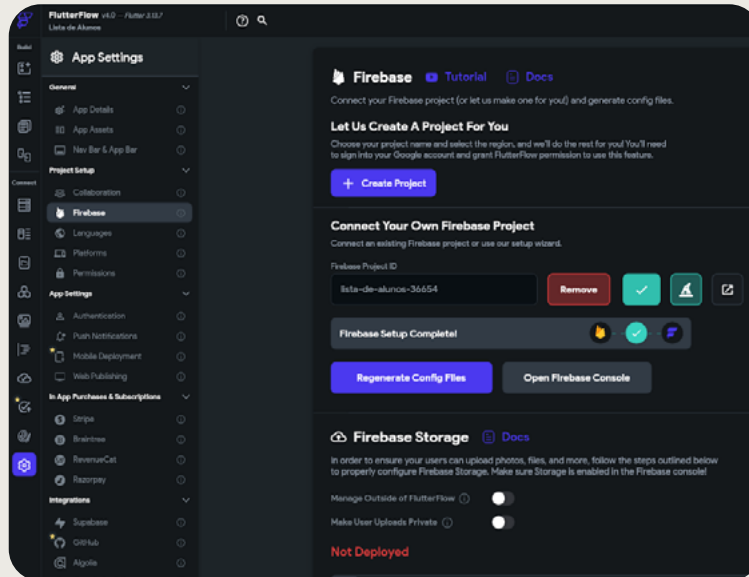
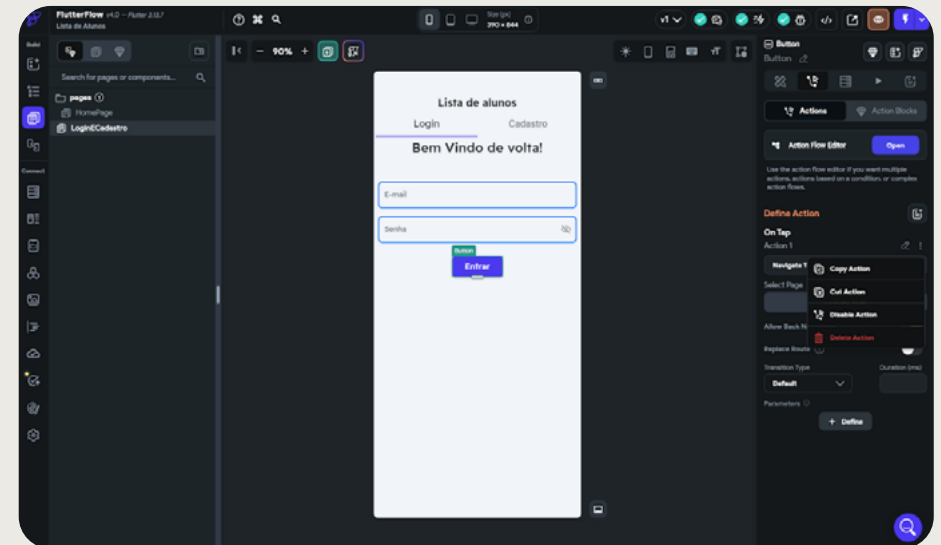
Devemos agora configurar todo o ambiente na primeira opção “Authentication Type”. Vamos falar por onde estamos fazendo a autenticação, marque a opção firebase.

Perceba que ele está nos acusando de um alerta em amarelo, ele está nos avisando que o Firebase ainda não foi configurado, vamos clicar neste aviso e ver para onde ele nos direciona.

É aqui onde vamos unir de fato o firebase ao FlutterFlow.

Em Connect Your Own Firebase Project ele pede o ID do nosso projeto, para conseguirmos esse ID é bem simples, basta ir ao site do firebase onde você criou o projeto, clicar na engrenagem e escolher a opção Configurações do projeto.

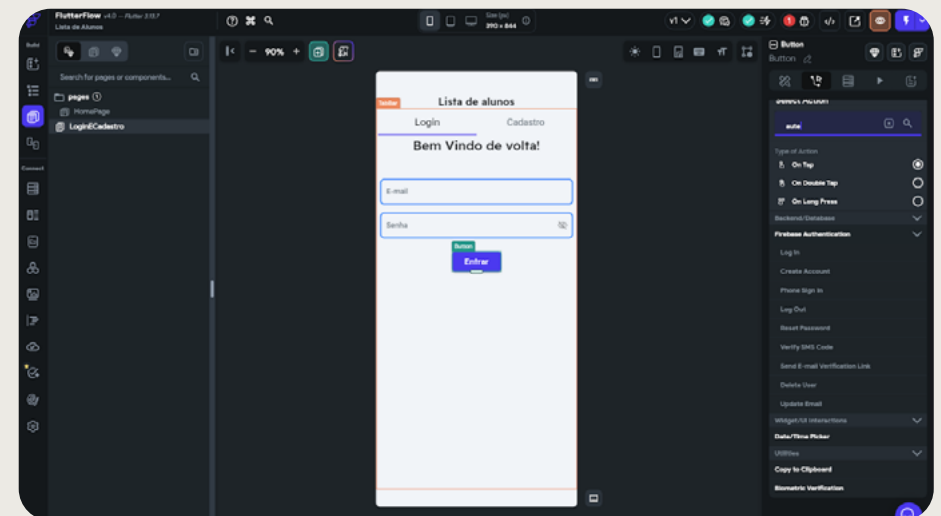
Abrirá uma nova página.

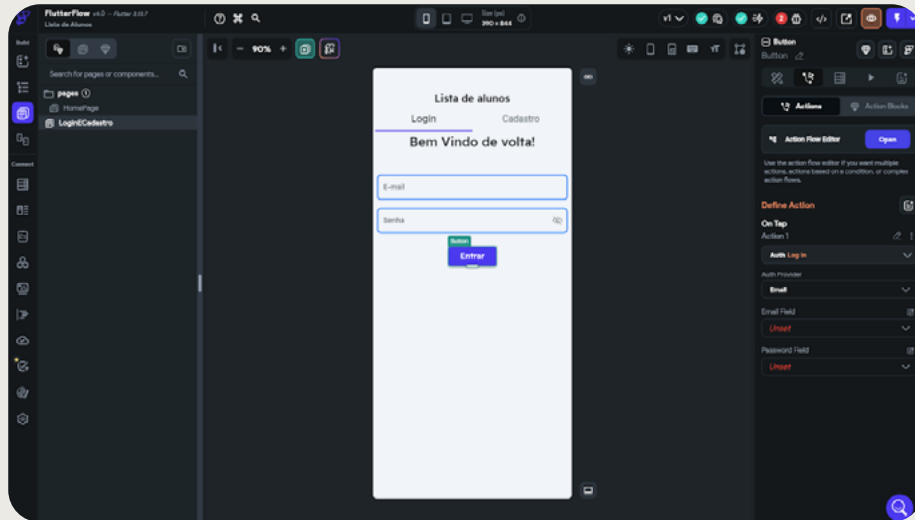
O nosso ID é o Código do projeto, copie ele e cole lá no Firebase project ID e clique em connect.

Uma mensagem de erro aparecerá.

Esse erro está falando que o FlutterFlow ainda não tem acesso a conta no firebase, portanto ele pede para adicionar o e-mail firebase@flutterflow.io como editor do projeto.

Vamos novamente na conta do firebase e clique em usuários e permissões.

Clique em adicionar membro.

Cole o e-mail que você copiou e dê o papel a ele como editor, por fim adicione o membro.

Agora, clique nas configurações avançadas de permissões e clique o ícone de lápis no e-mail que você acabou de adicionar.

Vamos adicionar mais dois papéis: o administrador do cloud functions e o usuário da conta de serviço e salvar.

Agora vamos em Authentication e clicar em começar.

Na página que aparece vamos habilitar a nossa forma de login, no caso será e-mail e senha.

Habilite a primeira opção e clique em salvar.

Pronto, está tudo devidamente configurado, vamos voltar para o FlutterFlow e tentar adicionar novamente o ID do projeto.

Sucesso! Conseguimos conectar, agora vamos clicar em Auto Generate Config Files para o FlutterFlow gerar todos os arquivos de configuração automaticamente.

Vamos excluir a ação de navigate que criamos e adicionar uma nova ação, clique no menu de opções e em deletar ação.

Vamos criar uma nova ação, na barra de pesquisa pesquise por “auto” e escolha no Firebase Authentication a opção log in.

Clique em Auth Provider e escolha a opção e-mail.

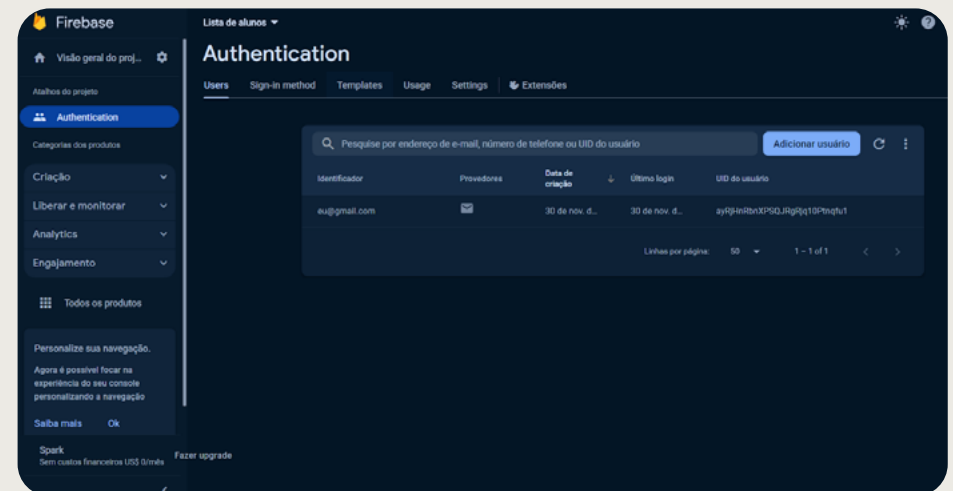
AGIR

Ótimo! Agora podemos referir os campos para gerar a autenticação, vamos para a nossa tela de login e cadastro.

Clique no botão Login.

Mostre ao FlutterFlow onde está o campo e-mail e o campo senha.

Faremos o mesmo procedimento no cadastro, porém não será log in, mas sim Create Account.

E vamos modificar os textos para e-mail, senha e confirmar senha, e não esqueça de definir o campo como um campo senha.

O login e cadastro já estarão funcionando corretamente, execute o projeto clicando no símbolo de raio para testar.

Ao cadastrar um e-mail ele irá automaticamente para a lista de usuários do firebase.

AULA 07

FlutterFlow

Materiais necessários:

Notebook, acesso à internet.

Duração:

2 horas

Objetivo:

Colocar ideias de aplicativos em prática.

INVESTIGAR

Peça para os alunos pesquisarem modelos de aplicativos e sites que eles têm ideias de fazer.

INVESTIGAR

Peça para os alunos colocarem todos os conhecimentos obtidos durante as aulas fazendo um aplicativo de ideias próprias.

ROBÓTICA COM
ARDUINO

COORDENADOR Marcos Vinicius Tavares Souza

AUTORES Tabatha Costa Bastos
Juliana da Silva Firmo
Tobias da Rodrigues Silva
Isaque Rodrigues Rocha
Vitória Sales de Jesus
Kawan Lucas Silva dos Santos
Mirelly Araújo Dos Santos
Renata Santos Celestino
Josué Pereira Da Crus
Marcos Vinicius Tavares Souza

REVISÃO Mariana Carolina de Almeida Souza

PROJETO GRÁFICO Patricia Karin S. Faber

MÓDULO 1

ROBÓTICO COM ARDUINO

AULA 01

Componentes e dispositivos eletrônicos

Materiais necessários:
Notebook, internet, resistor, jumpers, LED e protoboard.

Duração:
1 hora e 30 minutos

Objetivo:
Introduzir conceitos sobre resistores, jumpers, LED e protoboard.

ENGAJAR

1. Contextualizar

Dispositivos eletrônicos são pequenos componentes elétricos utilizados em conjuntos para realizar funções específicas. Esses dispositivos são comumente presentes em nossa vida, aprimorando e facilitando diversas tarefas. Exemplos de componentes e dispositivos eletrônicos que compõem sistemas incluem resistores, LEDs, protoboards e jumpers. Eles desempenham papéis cruciais em aplicações diversas, desde iluminação até a construção de circuitos eletrônicos mais complexos.

Resistores

Os resistores são componentes eletrônicos utilizados para controlar o fluxo de corrente elétrica. Cada resistor é projetado para ter uma resistência elétrica específica, que é medida em ohms (Ω).

A energia tende sempre procurar um caminho mais fácil para passar, imagine então que um resistor é como se fosse uma barreira que dificulta a passagem da energia. Dessa forma, quanto maior a resistência, menor será a corrente que passa pelo resistor.

Os resistores têm em seu corpo uma sequência de linhas coloridas, essas linhas determinam o valor de sua resistência elétrica. No código de cores, as primeiras linhas indicam sempre os primeiros dígitos do valor da resistência, a penúltima linha vai nos indicar o valor que iremos multiplicar pelos primeiros dígitos encontrados. Já a última linha indicará o máximo desvio do valor da resistência encontrada.

1.2 Ω 10%

Preto	0	0	1	
Marrom	1	1	10	1%
Vermelho	2	2	100	2%
Laranja	3	3	1 K	
Amarelo	4	4	10 K	
Verde	5	5	100 K	0.5%
Azul	6	6	1 M	0.25%
Violeta	7	7	10 M	0.1%
Cinza	8	8		0.05%
Branco	9	9		
Dourado			0.1	5%
Prata			0.01	10%

Fonte: https://mundoeducacao.uol.com.br/fisica/resistores.htm

É possível observar que o resistor na tabela tem apenas 4 faixas. É importante observar também que sempre teremos uma faixa mais afastada das outras (nesse caso é a faixa prata). A faixa mais afastada é a última, enquanto as mais próximas são as primeiras. Assim temos as cores marrom, vermelho, dourado e prata. Sabendo que a última e penúltima linhas são os valores do multiplicador e da tolerância da resistência, temos que o valor do nosso resistor (medido através da linha marrom e vermelha) é 12, como nossa terceira linha (dourada) nos dá um valor de multiplicador de 0,1 temos que: Resistência = 12 x 0,1 = 1,2Ω e sabendo que a última faixa é prata, sabemos que o valor da resistência varia em 10% de 1,2.

LED

Os LEDs são pequenos dispositivos eletrônicos que emitem luz quando uma corrente elétrica passa por eles. São bastante utilizados como fontes de luz em dispositivos eletrônicos. As cores dos LEDs variam de vermelho, amarelo, verde e azul, e dependem dos materiais semicondutores usados.

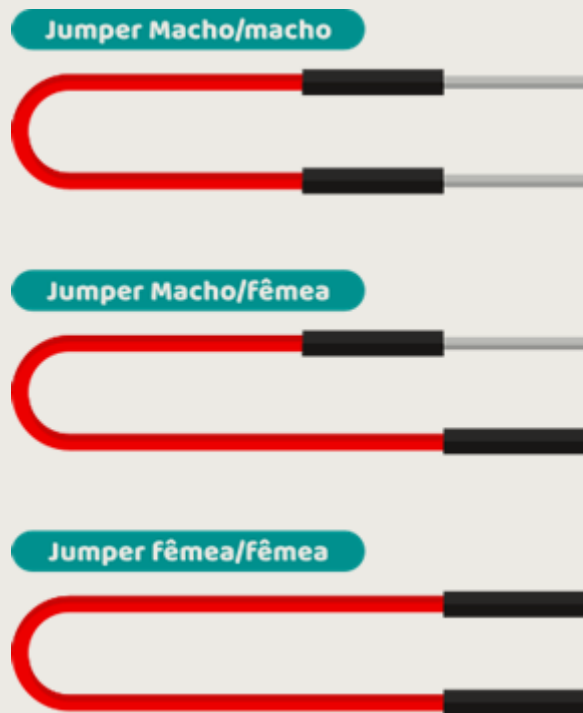
Jumpers

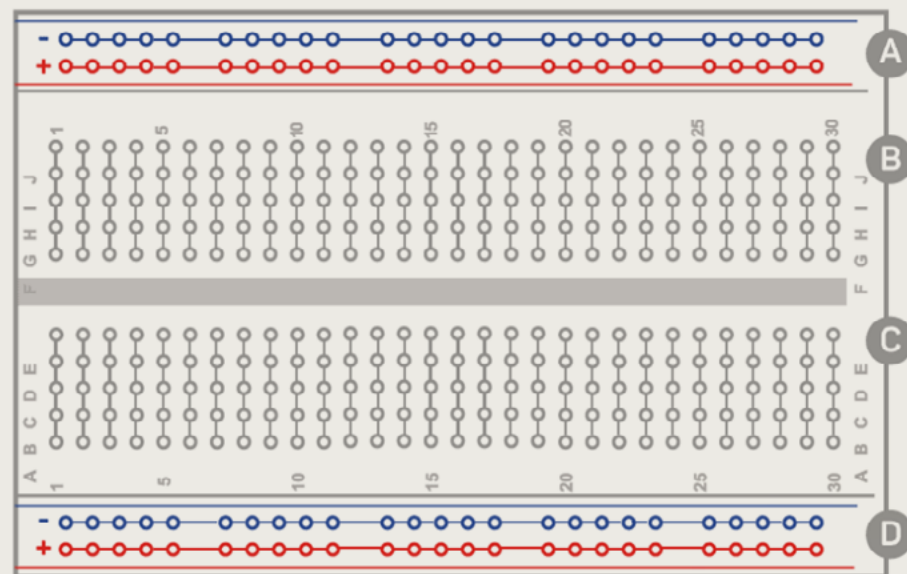
Jumpers são conectores usados em eletrônica para fazer conexões temporárias entre dois pontos em uma placa de circuito de forma prática. Eles podem ser do tipo macho/macho, macho/fêmea ou fêmea/fêmea.

Jumper Macho/macho

Jumper Macho/fêmea

Jumper fêmea/fêmea

Fonte: https://www.kitsarduino.com.br/cmp/jumpers.html

Protoboard

A protoboard, também conhecida como placa de prototipagem, é um componente fundamental para a montagem de circuitos eletrônicos temporários. Ela permite que você conecte e teste diversos componentes sem a necessidade de soldagem, tornando-a uma ferramenta valiosa no desenvolvimento de projetos de robótica com Arduino. Na protoboard os contatos em cada fila (linha) estão eletricamente conectados, bem como os contatos em cada coluna. Isso permite que você crie conexões elétricas facilmente, ligando os componentes aos pontos desejados.

INVESTIGAR

Multímetro

Atividade de pesquisa e fixação

Agora vamos fazer um exercício. Além dos dispositivos e componentes citados acima, o multímetro é um instrumento muito utilizado na eletrônica. Mas para que serve e como utilizar um multímetro?

Separe os alunos em pequenos grupos e peça para que os alunos pesquisem, em até 15 minutos, o que é um multímetro e como ele funciona. Eles podem pesquisar por vídeo, sites, documentos. Depois peça para que os alunos apresentem o que eles encontraram em sua busca em, no máximo, 2 minutos.

Exemplo de explicação

"Os Multímetros são instrumentos de medição que combinam diversas funções em um único dispositivo. São usados para medir grandezas elétricas como tensão, corrente e resistência em circuitos elétricos. Podem ser analógicos ou digitais e são ferramentas essenciais no mundo da eletrônica."

AGIR

Utilizar o multímetro

Peça para que os alunos utilizem algumas formas diferentes de encontrar a resistência de alguns resistores (eles podem utilizar alguns sites e tabelas para realizar essa tarefa). Os links abaixo mostram dois sites que podem ser utilizados para isso. Peça para que eles anotem os valores encontrados.

https://br.mouser.com/technical-resources/conversion-calculators/resistor-color-code-calculator

http://www.audioacustica.com.br/exemplos/Valores_Resistores/Calculadora_Ohms_Resistor.html

Peça para que os alunos utilizem o multímetro para verificar a resistência dos mesmos resistores e compararem com os valores encontrados anteriormente.

AULA 02

Arduino e IDE

Tema:

Componentes e dispositivos eletrônicos parte 2

Materiais necessários:

Notebook, internet, Arduino, resistor, jumpers, LED.

Duração:

1 hora e 30 minutos

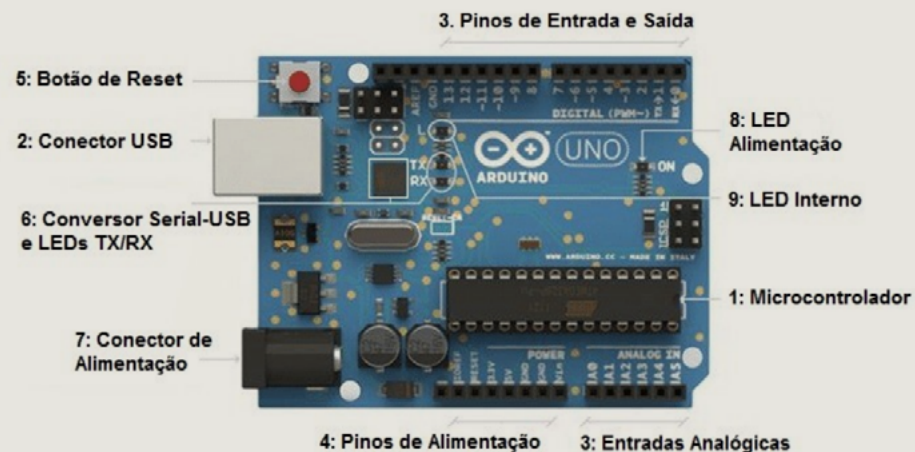
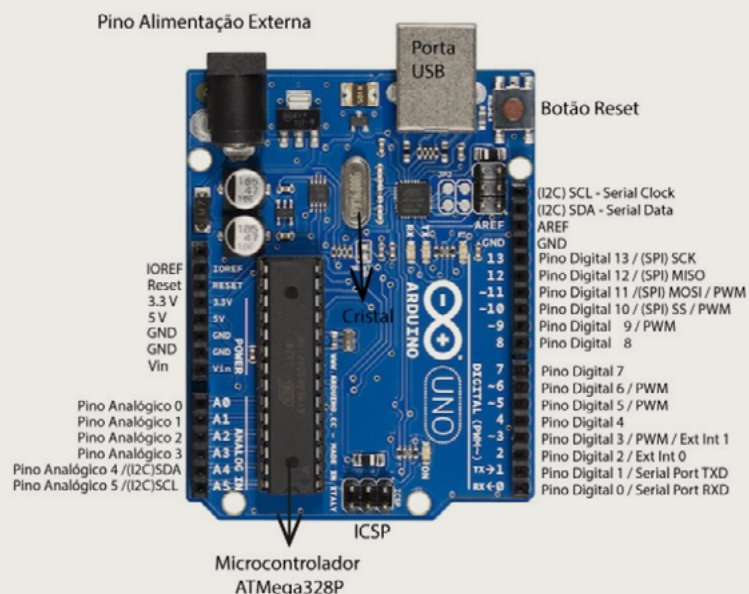
Objetivo:

Introduzir conceitos sobre Arduino, resistor, jumpers, LED e protoboard.

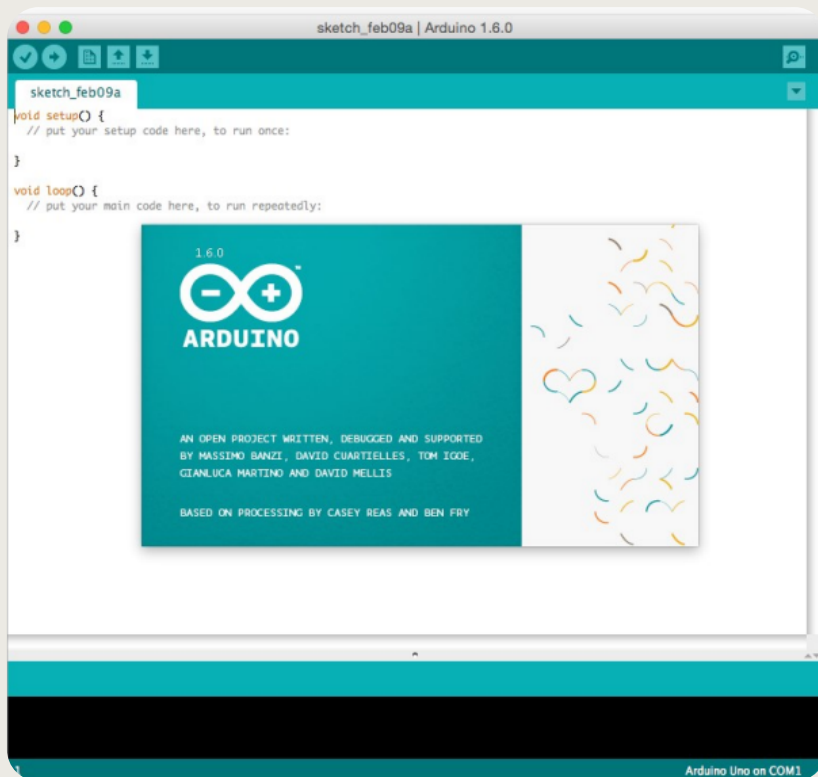
ENGAJAR

Arduíno Uno

O Arduino Uno é um tipo de placa microcontrolada baseada no chip ATmega328. O Arduino Uno é conhecido como a melhor placa para começar no desenvolvimento de projetos embarcados, sendo esta a mais usada e documentada de toda a família Arduino.

A placa Arduino é equipada com um microcontrolador, que é um chip capaz de processar instruções, controlar entradas, e saídas e executar tarefas específicas.

Detalhando alguns pinos

- **3,3V:**
  Pino de alimentação de 3,3V para os elementos dos circuitos.
- **5V:**
  Pino de alimentação de 5V para os elementos dos circuitos.
- **GND:**
  Ground ou pino terra, referência como voltagem 0V para os circuitos.
- **Portas digitais:**
  Entradas/saídas numeradas de 0 a 13, servem como entrada ou saída de sinais digitais para os sensores ou atuadores.
- **Portas analógicas:**
  Numeradas de 0 a 5, servem como entrada de dados vindos de sensores analógicos.

Software Arduino IDE

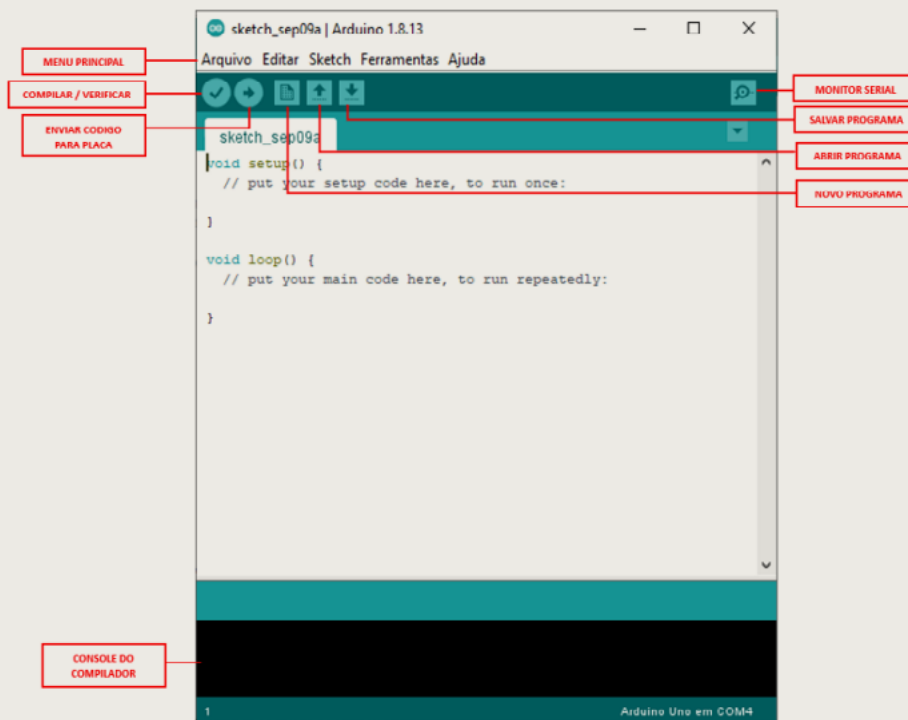
O Arduino IDE é um programa que você instala no seu computador para criar e carregar códigos nos seus projetos com Arduino. Ele é como um "escritório" onde você escreve instruções que os componentes do seu projeto vão seguir.

MENU PRINCIPAL:

No canto superior esquerdo, você encontrará o menu principal, que oferece opções para abrir, salvar e compilar projetos, bem como para carregar o código na placa Arduino. Aqui você também encontra opções para selecionar o tipo de placa, a porta serial e outras configurações relacionadas à programação da placa.

BOTÕES DE AÇÃO RÁPIDA:

Ao lado do menu principal, há uma série de botões de ação rápida que executam funções comuns, como criar um novo arquivo, abrir um arquivo existente, salvar, verificar (compilar) o código e carregar o código na placa.

ÁREA DE MENSAGENS (CONSOLE):

Abaixo da barra de tarefas, você verá uma área onde mensagens de status, erros de compilação e informações sobre a carga de código são exibidos.

Sintaxe básica da linguagem do Arduino

Função “setup()” (Configuração)

É executada uma vez quando o Arduino é ligado ou reiniciado. Ela é usada para realizar inicializações e configurações iniciais, como definir modos de pinos (entrada/saída) e iniciar comunicações.

Função “loop()” (Loop principal)

É onde a maior parte do código é executada. Ela é executada continuamente após o término da função “setup()”. Qualquer código colocado aqui será repetido em um loop infinito.

Comentários

Os comentários são trechos de texto no código que explicam o que cada parte do código faz. Eles são precedidos por “//” para comentários de uma linha ou entre “/*” e “*/” para comentários de várias linhas. Comentários ajudam a tornar o código mais compreensível para você e outros desenvolvedores.

Bibliotecas

Caso o projeto necessite de funcionalidades específicas, você pode incluir bibliotecas usando a diretiva “#include”. As bibliotecas fornecem funções e recursos pré-definidos que podem ser usados no seu código.

Variáveis

As variáveis são como “caixas” onde você guarda informações para usar depois. Imagine que você tem uma caixa chamada “idade” e nela você coloca a sua idade. No código, podemos criar variáveis para armazenar números, textos e outros dados.

Tipos de Dados

Os tipos de dados determinam que tipo de informação uma variável pode guardar. São como os diferentes tipos de caixas para coisas diferentes. Alguns tipos comuns no Arduino são: int (números inteiros), float (números com casas decimais), char (caracteres individuais) e string (texto).

Operadores

Os operadores são símbolos especiais que nos ajudam a fazer coisas com as variáveis. São como os sinais de +, -, * e / em.

INVESTIGAR

Atividade de pesquisa e fixação

Vamos fazer um exercício. Para que possamos iniciar as nossas aulas práticas, temos primeiramente que instalar o software Arduino IDE.

Atividade: Pesquisar e instalar o software Arduino IDE.

10 minutos para os alunos realizarem a atividade.

Possível resposta:

Download e instalação.

Visite o site oficial do Arduino (https://www.arduino.cc) e vá para a seção “Software” ou “Software > Downloads”.

Escolha a versão do Arduino IDE compatível com o seu sistema operacional (Windows, macOS ou Linux).

Faça o download do instalador e siga as instruções para instalá-lo no seu computador.

AGIR

Desafio:

Adicionar uma biblioteca ao software Arduino IDE.

Possível resposta:

- Abra o Arduino IDE;
- No menu superior localize a opção “Sketch”;
- Procure a opção “Include Library”;
- Selecione a opção “Manage Libraries...”;
- Busque pela biblioteca que deseja instalar;
- Selecione a versão desejada;
- Clique em “Install”.

Se o processo ocorreu de forma correta, uma tela preta aparecerá na parte inferior da tela com a mensagem de confirmação.

AULA 03

Acendendo um Led

Materiais necessários:

Notebook, internet, Arduino, resistor, jumpers e LED.

Duração:

1 hora e 30 minutos

Objetivo:

Contextualizar e mostrar na prática as diversas aplicações e usos de um LED.

ENGAJAR

1. Contextualização/engajar

Os LEDs são pequenos dispositivos que convertem energia elétrica diretamente em luz, quando uma corrente elétrica passa por eles. Uma característica dos LEDs é sua eficiência na conversão de energia elétrica em luz, gerando menos calor do que outras fontes de luz convencionais, como lâmpadas incandescentes ou fluorescentes. Tipos de **LEDs:**

- **LEDs de alto brilho:** Projetados para aplicações de iluminação de alta intensidade. As cores dos LEDs variam de vermelho, amarelo, verde e azul e dependem dos materiais semicondutores usados.
- **LEDs RGB:** Utilizam LEDs vermelhos, verdes e azuis combinados para produzir uma gama mais ampla de cores.
- **OLEDs (Organic Light Emitting Diodes):** Utilizam materiais orgânicos e oferecem flexibilidade em aplicações de tela.

O LED tem sempre uma "perninha" menor que a outra, e isso é o que diferencia sua polaridade, ou seja, a perninha menor é a negativa (cátodo) e a maior é a positiva (ânodo).

+ Anodo
- Catodo

Fonte: https://josecintra.com/blog/led-arduino-controle/

2. Apresentação de exemplos e referências

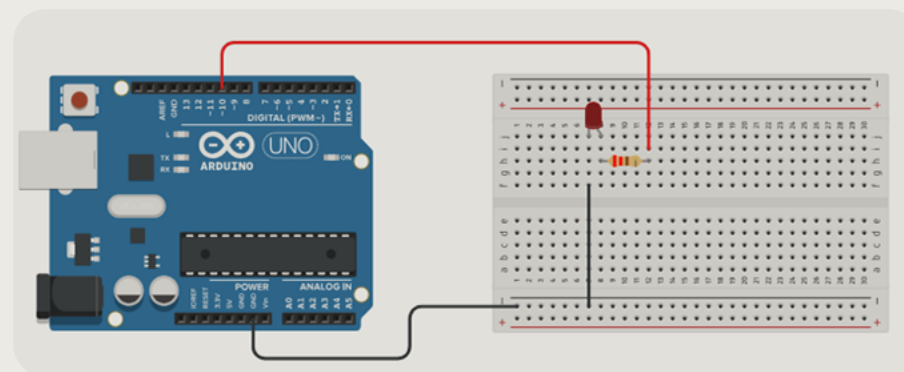
Objetivo:

Essa prática consiste em ligar e desligar o LED (ou seja, piscar o LED) em tempo definido no código.

Materiais necessários:

- Uma protoboard;
- Um LED;
- Três jumpers M-M;
- Um resistor de 220 ohms;
- Um cabo USB;
- Um computador com a IDE do Arduino instalada;
- Um microcontrolador Arduino.

Montagem:

UNO
ARDUINO
DIGITAL (PWM~)
POWER
ANALOG IN

Código:

```
void setup() {
    pinMode(10, OUTPUT);
}
void loop() {
    digitalWrite(10, HIGH);
    delay(1000);
    digitalWrite(10, LOW);
    delay(1000);
}
```

INVESTIGAR

3. Atividade de pesquisa e fixação

Agora vamos fazer um exercício. Além das áreas citadas, quais outros exemplos de uso e aplicação de LEDs estão presentes no nosso dia a dia?

Atividade: Pesquisar aplicações e usos do LED.

10 minutos para os alunos realizarem a pesquisa.

Necessário destacar a importância de pesquisa de forma adequada, no caso dos LEDs, a depender de como seja realizada a pesquisa, é possível encontrar respostas como diversas, como "Uso para o LED no dia a dia", "Aplicações dos LEDs", dentre outros.

Possíveis respostas:

- **Iluminação residencial e pública:**
  Lâmpadas de LED são amplamente usadas para iluminação interna e externa em residências, escritórios, lojas e estabelecimentos comerciais, postes de luz, iluminação de ruas e fachadas de edifícios. As lâmpadas de LED consomem menos energia e têm uma vida útil mais longa em comparação com as lâmpadas tradicionais.

Fonte: https://jorgezanoni.com.br/2019/projeto-para-modernizacao-da-iluminacao-publica-de-dracena-ja-foi-aprovado-e-assinado-na-prefeitura/

Eletrônicos:
A maioria dos dispositivos móveis, como smartphones, tablets e laptops, utilizam, LEDs em suas telas, proporcionando brilho e nitidez nas imagens.

Sinalização:
Os LEDs são amplamente utilizados em sinais de trânsito, luzes de emergência, placas publicitárias e displays de informações públicas, devido à sua visibilidade e eficiência.

MSG Sphere
Esfera em leds utilizada para realizar anúncios e projetar imagens.

Fonte: https://www.sportindustry.biz/news-categories/cutting-edge-sport/unique-sphere-sports-venue-launches-in-las-vegas/

Iluminação de veículos:
Faróis, lanternas traseiras e luzes indicadoras em veículos modernos são frequentemente equipados com LEDs devido à sua eficiência.

Setor agrícola (Cultivo Indoor):
Os LEDs são empregados em fazendas verticais ou sistemas de cultivo indoor, fornecendo luzes adequadas para o crescimento de plantas, permitindo um controle preciso do espectro luminoso para diferentes estágios de crescimento.

Aplicações médicas:
São usados em equipamentos médicos para fototerapia, iluminação cirúrgica e até mesmo em terapias de luz para estimular o crescimento celular ou tratar condições específicas da pele.

Iluminação pública e urbana:
Muitas cidades utilizam LEDs em postes de luz, semáforos e sinalizações devido à sua eficiência energética, economia de custos a longo prazo e capacidade de iluminar áreas maiores.

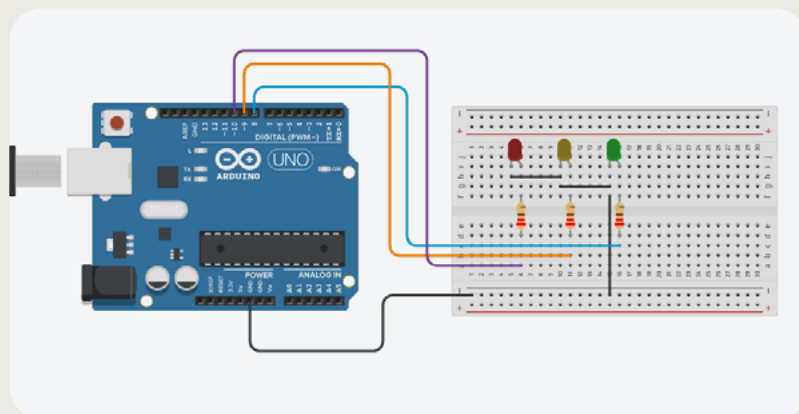
AGIR

Desafio:

Objetivo: Realizar a montagem que simule um semáforo, alternando entre as três cores (verde, amarelo e vermelho).

Materiais necessários:

- Uma protoboard;
- Um LED vermelho;
- Um LED amarelo;
- Um LED verde;
- Sete jumpers M-M;
- Três resistores de 220 ohms;
- Um cabo USB;
- Um computador com a IDE do Arduino instalada;
- Um microcontrolador Arduino.

Possível resposta:

Fonte: Equipe Ed-mundo

Código:

```
void setup() {
        pinMode(8, OUTPUT);     //define o pino 8 como saída para o LED verde
        pinMode(9, OUTPUT);     //define o pino 9 como saída para o LED amarelo
        pinMode(10, OUTPUT);    //define o pino 10 como saída para o LED vermelho
}
void loop() {
        digitalWrite(8, HIGH);
        delay(4000);
        digitalWrite(8, LOW);
        digitalWrite(9, HIGH);
        delay(2000);
        digitalWrite(9, LOW);
        digitalWrite(10, HIGH);
        delay(4000);
        digitalWrite(10, LOW);
}
```

AULA 04

Utilizando o PUSHBUTTON

Tema:
PUSHBUTTON

Materiais necessários:
Notebook, internet, Arduino, resistor, jumpers, pushbutton e LED.

Duração:
1 hora e 30 minutos

Objetivo:
Entender o funcionamento de um pushbutton (botão).

ENGAJAR

1. Contextualização/engajar

Os pushbuttons, ou botões de pressão, são componentes simples e comuns na eletrônica. Eles consistem em um interruptor momentâneo que é ativado quando pressionado e desativado quando liberado. Sua estrutura básica é composta por duas partes condutoras separadas por um isolante. Quando pressionado, as partes condutoras se conectam, permitindo o fluxo de corrente elétrica.

Fonte: https://www.eletrogate.com/push-button-chave-tactil-6x6x6mm

Os pushbuttons são uma das formas mais básicas e intuitivas de interação entre humanos e dispositivos eletrônicos. São encontrados em muitos dispositivos do dia a dia, como controles remotos, teclados, joysticks e muitos outros.

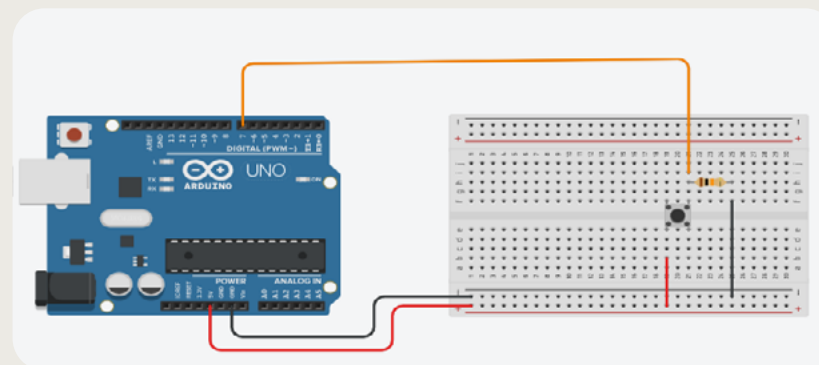
2. Apresentação de exemplos e referências

Objetivo:

Essa prática consiste em controlar o funcionamento de um LED, este deve ligar e desligar ao pressionar o pushbutton (botão).

Material necessário:

- Uma protoboard;
- Um LED;
- Sete jumpers M-M;
- Um resistor de 220 ohms;
- Um resistor de 10K ohms;
- Um pushbutton;
- Um cabo USB;
- Um computador com a IDE do Arduino instalada;
- Um microcontrolador Arduino.

Possível resposta:

Fonte: Equipe Ed-mundo

Código:

```
const byte BOTAO = 7;

byte valorBotao;

byte valorBotaoAnterior;

int cont = 0;

void setup(){

 pinMode(BOTAO, INPUT);

 Serial.begin(9600);

}

void loop(){

 valorBotao = digitalRead(BOTAO);

 if (valorBotao != valorBotaoAnterior){

  if (valorBotao){

 cont = cont + 1;

  }

 }

 valorBotaoAnterior = valorBotao;

 Serial.println(cont);

 delay(10);

}
```

AULA 05

Sensores - Parte 1

Materiais necessários:
Notebook, internet, sensor LDR, Arduino, protoboard, jumpers, LED e resistores.

Duração:
1 hora e 30 minutos

Objetivo:
Contextualizar e mostrar na prática as diversas aplicações e usos dos sensores.

ENGAJAR

1. Contextualizar

Os sensores são dispositivos que têm a capacidade de perceber e capturar informações do mundo físico ao seu redor, convertendo essas informações em sinais elétricos ou digitais que podem ser processados por sistemas eletrônicos. Eles desempenham um papel crucial em inúmeros campos, desde a automação industrial até a tecnologia residencial, monitoramento ambiental e saúde.

A conexão de sensores com o Arduino pode variar dependendo do tipo de sensor utilizado, mas há etapas básicas que são comuns para diversos sensores:

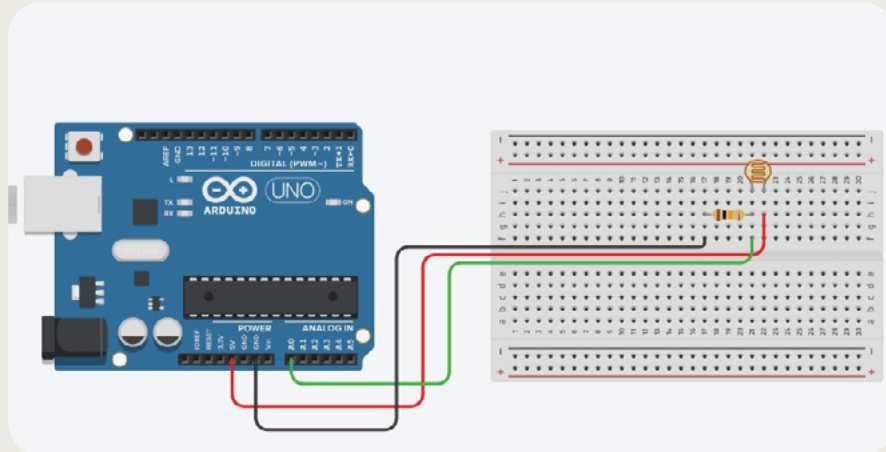
- **Identificação dos pinos do sensor:** Cada sensor possui pinos específicos para alimentação, terra (GND) e sinais de saída. Identificar esses pinos é o primeiro passo.
- **Conexão de alimentação (VCC) e terra (GND):** Conecte o pino de alimentação (VCC ou +) do sensor a uma porta de alimentação (5V ou 3.3V) do Arduino. E conecte o pino terra (GND) do sensor a um dos pinos GND (terra) do Arduino para fechar o circuito.
- **Conexão do pino de saída do sensor:** Conecte o pino de saída do sensor a um dos pinos de entrada do Arduino. Dependendo do sensor, essa conexão pode ser analógica (A0 a A5) ou digital (0 a 13).
- **Uso de resistores (quando necessário):** Alguns sensores requerem resistores para garantir uma leitura correta.
- **Programação no Arduino:** Escreva um código no Arduino IDE para ler os dados do sensor. O código pode variar de acordo com o tipo de sensor e a informação que se deseja obter. Por exemplo, para um sensor de luminosidade, o código pode envolver a leitura analógica de um valor que representa a intensidade da luz.

2. Apresentação de exemplos e referências

Objetivo: Essa prática consiste em coletar a luminosidade do ambiente em intervalo definido no código.

Materiais necessários:

- Uma protoboard;
- Um sensor LDR;
- Três jumpers M-M;
- Um resistor de 10K ohms;
- Um cabo USB;
- Um computador com a IDE do Arduino instalada;
- Um microcontrolador Arduino.

Montagem:

Fonte: Equipe Ed-mundo

Código:

```
int ldr = A0;
int valorldr = 0;
void setup() {
  pinMode(ldr, INPUT);
  Serial.begin(9600);
}
void loop() {
  valorldr = analogRead(ldr);
  Serial.println(valorldr);
  delay(2000);
}
```


Atividade de pesquisa e fixação

Agora vamos fazer um exercício. Além do sensor LDR utilizado na explicação, que outros sensores podemos encontrar no nosso dia a dia?

Atividade: Pesquisar sensores presentes no nosso dia a dia.

10 minutos para os alunos realizarem a pesquisa.

Possíveis respostas:

- **Acelerômetro:** Presente em smartphones e dispositivos portáteis para detecção de movimento e orientação.
- **Giroscópio:** Também comum em smartphones e dispositivos de realidade virtual, ajuda na detecção de rotação e movimentos angulares.
- **GPS:** Utilizado em smartphones, veículos e dispositivos de navegação para determinar a localização geográfica.
- **Sensor de proximidade:** Presente em smartphones e dispositivos de detecção de proximidade para desligar a tela durante uma chamada ou para interações de gestos.
- **Sensor de temperatura:** Usado em termostatos, eletrodomésticos, sistemas de climatização e até mesmo em wearables para monitorar a temperatura corporal.
- **Sensor de umidade:** Comum em medidores climáticos, sistemas de irrigação automatizados e dispositivos de controle de umidade.

- **Sensor de som (Microfone):** Presente em smartphones, dispositivos de IoT e sistemas de segurança para capturar e interpretar sinais sonoros.
- **Sensor de batimentos cardíacos:** Usado em smartwatches e dispositivos médicos para monitorar a frequência cardíaca.
- **Sensor de toque/capacitivo:** Encontrado em telas sensíveis ao toque, botões de elevadores e dispositivos de controle de acesso.
- **Sensor de pressão:** Utilizado em barômetros, altímetros e medidores de pressão atmosférica.
- **Sensor de gás:** Presente em detectores de monóxido de carbono, sistemas de segurança e controle de qualidade do ar.
- **Sensor de distância (Ultrassônico ou Infravermelho):** Encontrado em robôs, sistemas de estacionamento automático e drones para medir distâncias.

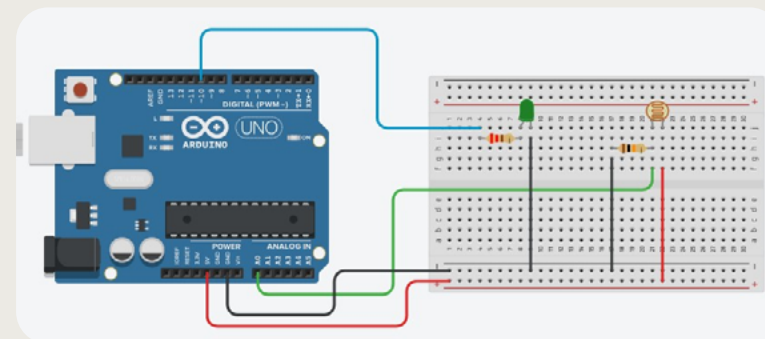
AGIR

Desafio:

Objetivo: Realizar a montagem "Iluminação Inteligente". Esse desafio consiste em coletar a luminosidade do ambiente e ligar o LED caso o valor seja menor que 750.

Materiais necessários:

- Uma protoboard;
- Sete jumpers M-M;
- Um LED;
- Um resistor de 220 ohms;
- Um resistor de 10K ohms;
- Um sensor LDR;
- Um cabo USB;
- Um computador com a IDE do Arduino instalada;
- Um microcontrolador Arduino.

Possível resposta:

Fonte: Equipe Ed-mundo

Código:

```
int led = 10;
int ldr = A0;
int valorldr = 0;
void setup() {
 pinMode(led, OUTPUT);
 pinMode(ldr, INPUT);
 Serial.begin(9600);
}
void loop() {
 valorldr = analogRead(ldr);
 Serial.println(valorldr);
 if ((valorldr) < 750) {
  digitalWrite(led, HIGH);
 }
 else {
  digitalWrite(led, LOW);
 }
}
```

AULA 06

Sensores - Parte 2

Materiais necessários:

Notebook, internet, sensor de temperatura TMP36, Arduino, protoboard e jumpers.

Duração:

1 hora e 30 minutos

Objetivo:

Contextualizar e mostrar na prática as diversas aplicações e usos dos sensores.

ENGAJAR

1. Contextualizar

Os sensores de temperatura são dispositivos eletrônicos projetados para medir e quantificar a temperatura ambiente ou de objetos específicos. Eles são utilizados em uma ampla gama de aplicações, desde sistemas de controle de climatização até dispositivos médicos e eletrônicos.

Apresentação de exemplos e referências

Os sensores de temperatura desempenham um papel crucial em várias aplicações. Temos agora alguns exemplos de sensores de temperatura comuns:

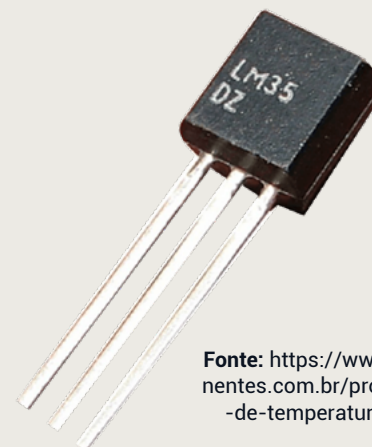
- **Termistores:** Sua resistência elétrica varia de maneira inversa ou direta com a temperatura, respectivamente.

Fonte: https://www.powercircuit.com.br/produtos/12030/12028/1/Termistor-NTC/TERMISTOR-NTC-10K-OHMS-2MM-P20MM

- **Termopares:** Consistem em dois fios de diferentes metais unidos em uma extremidade. A diferença de temperatura entre as extremidades gera uma voltagem que pode ser medida para calcular a temperatura.

Fonte: https://plimcan.com/aquecimento/termopar/

- **Sensores de semicondutor (como o LM35):** Baseados na variação da tensão de saída em relação à mudança de temperatura, geralmente apresentam uma leitura de temperatura linear.

Fonte: https://www.dedcomponentes.com.br/produto/sensor-de-temperatura-lm-35-dz

INVESTIGAR

Atividade de pesquisa e fixação

Vamos fazer um exercício. Em quais instrumentos do dia a dia podemos encontrar sensores de temperatura?

Atividade:

Pesquisar instrumentos do nosso dia a dia em que podemos encontrar sensores de temperatura.

10 minutos para os alunos realizarem a pesquisa.

Possíveis respostas:

- **Termômetros digitais e analógicos:** São usados para medir a temperatura corporal, ambiente ou de objetos específicos. Termômetros digitais geralmente utilizam sensores de temperatura para fazer medições precisas.
- **Aparelhos eletrônicos (laptops, celulares):** Possuem sensores internos para monitorar a temperatura do processador e evitar superaquecimento, além de garantir o funcionamento adequado do dispositivo.
- **Sistemas de climatização e ar-condicionado:** Utilizam sensores de temperatura para monitorar e regular a temperatura ambiente, ajustando o sistema conforme a configuração desejada.
- **Termômetros infravermelhos (pistolas de temperatura):** Encontrados em estabelecimentos comerciais, clínicas e até mesmo em casas, são usados para medir a temperatura à distância, sem contato direto.
- **Termômetros digitais e analógicos:** São usados para medir a temperatura corporal, ambiente ou de objetos específicos. Termômetros digitais geralmente utilizam sensores de temperatura para fazer medições precisas.
- **Eletrodomésticos:** Geladeiras, freezers e fogões podem conter sensores de temperatura para garantir que as condições internas estejam adequadas para armazenamento de alimentos e para controlar o aquecimento durante o cozimento.
- **Veículos automotivos:** Carros modernos possuem sensores de temperatura para monitorar o resfriamento do motor, regular o ar-condicionado e controlar sistemas de aquecimento interno.
- **Equipamentos médicos:** Desde incubadoras neonatais até equipamentos de ressonância magnética, muitos dispositivos médicos usam sensores de temperatura para monitorar e controlar as condições de temperatura.
- **Indústrias:** Em processos de fabricação industrial, há uma ampla utilização de sensores de temperatura para monitorar e controlar os processos, garantindo qualidade e eficiência na produção.

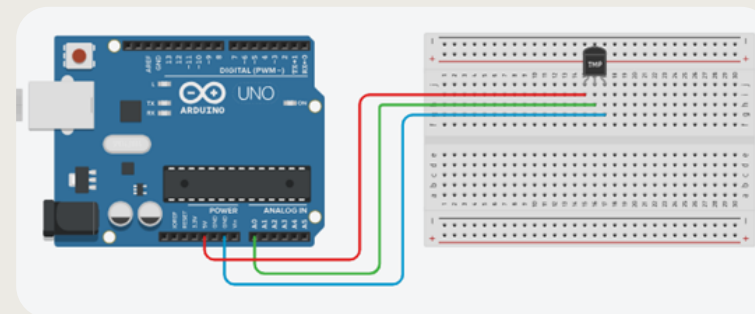
A
G
I
R

Desafio:

Objetivo: Essa prática consiste em coletar a temperatura do ambiente em intervalo definido no código e exibir no monitor serial.

Materiais necessários:

- Uma protoboard
- Um sensor de temperatura TMP36;
- Três jumpers M-M;
- Um cabo USB;
- Um computador com a IDE do Arduino instalada;
- Um microcontrolador Arduino.

Montagem:

Fonte: Equipe Ed-mundo

Código:

```
#include <Thermistor.h>
Thermistor temp(2);
void setup() {
 Serial.begin(9600);
 delay(1000);
}
void loop() {
 int temperature = temp.getTemp();
 Serial.print("Temperatura: ");
 Serial.print(temperature);
 Serial.println("*C");
 delay(1000);
}
```

AULA 07

Sensor ultrassônico

Materiais necessários:

Notebook, internet, sensor ultrassônico HC-SR04, Arduino, protoboard e jumpers.

Duração:

1 hora e 30 minutos

Objetivo:

Contextualizar e mostrar na prática as diversas aplicações e usos dos sensores.

ENGAJAR

Contextualizar

O potenciômetro é um componente fundamental na eletrônica utilizado para controlar a resistência elétrica em um circuito. Ele consiste em um resistor de três terminais, com uma conexão móvel que pode ser ajustada para alterar a resistência entre os terminais. Essa capacidade de variar a resistência elétrica o torna extremamente versátil em diferentes aplicações eletrônicas. Com papel crucial em diversas áreas, pode ser utilizado de diversas formas:

- **Controle de volume e brilho:** Em dispositivos como amplificadores de áudio ou equipamentos de vídeo.
- **Controle de velocidade e posição:** Em motores elétricos, servomecanismos ou robótica.
- **Calibração e ajustes fino:** Em instrumentos de medição, circuitos de precisão ou laboratórios.
- **Controle de tonais em instrumentos musicais:** Em guitarras, sintetizadores e outros instrumentos musicais, os potenciômetros podem ser utilizados para controlar tons, equalizadores e efeitos sonoros.

INVESTIGAR

Atividade de pesquisa e fixação

Vamos fazer uma atividade. Quais seriam as aplicações práticas para os LEDS e potenciômetro juntos?

Atividade:

Pesquisar e discutir com os colegas aplicações práticas que envolvam o uso conjunto de LEDs e potenciômetros.

10 minutos para os alunos realizarem a pesquisa.

Possíveis respostas:

- **Iluminação ajustável:** Usar potenciômetros para controlar a luminosidade de LEDs em lâmpadas ou sistemas de iluminação, permitindo ajustes personalizados de brilho.
- **Indicadores visuais ajustáveis:** Em painéis de controle ou dispositivos eletrônicos, LEDs podem ser usados como indicadores visuais de status, e potenciômetros podem controlar a intensidade luminosa para adaptação a diferentes condições de iluminação.
- **Arte e design:** Em projetos artísticos, os LEDs podem criar efeitos visuais diferentes. O uso de potenciômetros permite o controle sobre a intensidade da iluminação, proporcionando flexibilidade criativa.
- **Ambientes inteligentes:** Em sistemas de automação residencial ou comercial, os LEDs podem ser parte de um sistema de iluminação inteligente controlado por potenciômetros para ajustar a atmosfera ou a iluminação de acordo com as preferências do usuário.
- **Prototipagem de circuitos eletrônicos:** Em laboratórios ou para fins educacionais, essa combinação é útil para prototipar circuitos eletrônicos simples e aprender sobre o controle de luminosidade em circuitos.

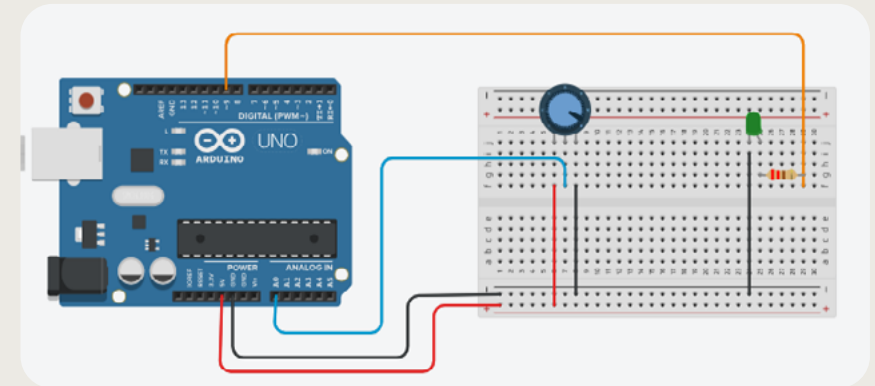
AGIR

Desafio:

Objetivo: Esse desafio consiste em controlar a intensidade do LED através do potenciômetro.

Materiais necessários:

- Uma protoboard;
- Um potenciômetro;
- Um LED;
- Um resistor de 220 ohms;
- Sete jumpers M-M;
- Um cabo USB;
- Um computador com a IDE do Arduino instalada;
- Um microcontrolador Arduino.

Montagem:

Fonte: Equipe Ed-mundo

Código:

```
int ledPin = 9;
int potPin = A0;
int valorpot = 0;
int pwm = 0;
void setup(){
 pinMode(ledPin, OUTPUT);
 pinMode(potPin, INPUT);
 Serial.begin(9600);
}
void loop(){
 valorpot = analogRead(potPin);
 pwm = map(valorpot, 0, 1023, 0, 255);
 Serial.println(pwm);
 analogWrite(ledPin, pwm);
 delay(500);
 }
```

AULA 09

LED RGB

Tema:
Cores

Materiais necessários:
Notebook, internet, LED RGB, resistores, Arduino, protoboard e jumpers.

Duração:
1 hora e 30 minutos

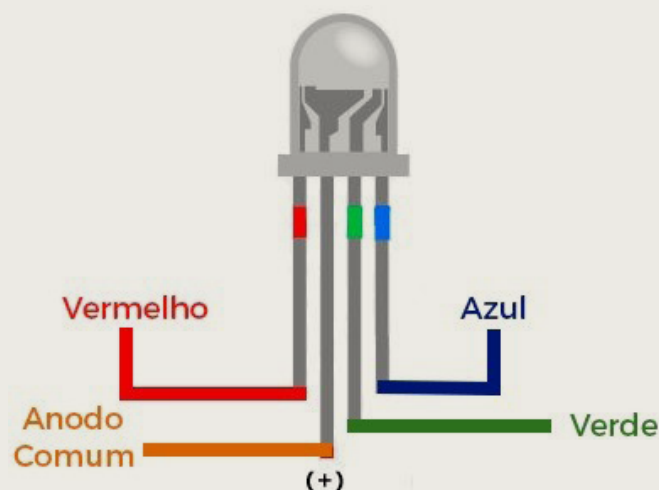
Objetivo:
Contextualizar as cores secundárias e terciárias por meio da tecnologia.

ENGAJAR

Contextualizar

Os LEDs RGB são dispositivos semicondutores que emitem luz em três cores primárias (vermelho, verde e azul), permitindo uma vasta gama de cores quando combinadas em diferentes intensidades. A vantagem desses LEDs é a flexibilidade e controle sobre a luz emitida, o que os torna muito populares em uma ampla variedade de aplicações.

Vermelho
Azul
Anodo Comum
Verde
(+)

Fonte: https://portal.vidadesilicio.com.br/como-utilizar--o-led-rgb-com-arduino/

INVESTIGAR

Atividade de pesquisa e fixação

Vamos fazer uma atividade. Em que momentos podemos aplicar os LEDs RGB?

Atividade:
Pesquisar aplicações comuns dos LEDs RGB.

10 minutos para os alunos realizarem a pesquisa.

Possíveis respostas:

- **Iluminação decorativa:** Em sistemas de iluminação decorativa para residências, escritórios, palcos de shows, festas e eventos, onde a capacidade de alterar a cor da luz é desejada para criar ambientes personalizados.
- **Sinalização e displays:** Nos setores de sinalização e publicidade, os LEDs RGB são utilizados para criar displays de alta resolução, sinalização dinâmica e painéis de informações que podem exibir texto, gráficos e vídeos coloridos.
- **Entretenimento e tecnologia audiovisual:** Em televisões, monitores e sistemas de entretenimento para proporcionar uma experiência visual mais rica, com imagens mais vibrantes e coloridas.
- **Iluminação automotiva:** Em sistemas de iluminação de veículos, como faróis adaptativos ou iluminação interna, oferecendo opções de cores e efeitos visuais.
- **Prototipagem eletrônica e projetos DIY:** Nos projetos de eletrônica, os LEDs RGB são utilizados para criar sistemas de iluminação personalizados, protótipos interativos e experimentações em eletrônica criativa.

AGIR

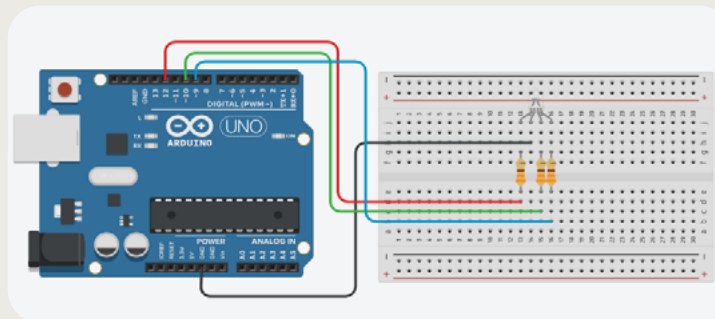
Desafio:

Objetivo: Esse desafio consiste em criar efeitos de iluminação dinâmicos com transições de cores e sequências pré-definidas.

Proposta de solução

Materiais necessários:

Uma protoboard;
Um LED RGB;
Quatro jumpers M-M;
Três resistores de 330 ohms;
Um cabo USB;
Um computador com a IDE do Arduino instalada;
Um microcontrolador Arduino.

Montagem:

Fonte: Equipe Ed-mundo

Código:

```
void setup()
{
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(12, OUTPUT);
}
void loop()
{
  digitalWrite(9, HIGH);
  digitalWrite(10, HIGH);
  delay(2000);
  digitalWrite(9, LOW);
  digitalWrite(10, LOW);
  delay(2000);
  digitalWrite(12, HIGH);
  digitalWrite(9, HIGH);
  delay(2000);
  digitalWrite(12, LOW);
  digitalWrite(9, LOW);
  delay(2000);
}
```

AULA 10

Placa Solar

Tema:
Sensores

Materiais necessários:
Notebook, internet, placa solar, LED, protoboard e jumpers.

Duração:
1 hora e 30 minutos

Objetivo:
Contextualizar e mostrar na prática as diversas aplicações e usos dos sensores.

ENGAJAR

Contextualizar

As placas solares, também conhecidas como painéis fotovoltaicos, são dispositivos semicondutores que convertem a luz solar diretamente em energia elétrica por meio do chamado "efeito fotovoltaico". Essas placas são componentes fundamentais dos sistemas de energia solar, sendo parte essencial da tecnologia de captação de energia solar.

Fonte: https://www.eletronicasystem.com.br/produto/mini-placa-painel-solar-12v-150ma-11x11cm-arduino-robotica-03-363

Tipos de placas solares:

- **Monocristalinas:** São feitas de silício monocristalino, possuindo uma eficiência relativamente alta na conversão de luz solar em eletricidade. Geralmente são mais eficientes, mas também mais caras de produzir.
- **Policristalinas:** Feitas de silício policristalino, são menos eficientes que as monocristalinas, porém são mais econômicas de fabricar.
- **Filmes finos:** Utilizam materiais semicondutores em forma de finas camadas depositadas sobre um substrato, sendo menos eficientes, mas mais flexíveis e leves.

INVESTIGAR

Atividade de pesquisa e fixação

Vamos fazer um exercício. Quais são as aplicações comuns para as placas solares?

Atividade: Pesquisar e discutir com os colegas aplicações comuns para as placas solares.

10 minutos para os alunos realizarem a pesquisa.

Possíveis respostas:

- **Geração de energia residencial e comercial:** São usadas para gerar energia elétrica em residências, edifícios comerciais e indústrias, reduzindo a dependência da rede elétrica convencional e permitindo a geração própria de energia limpa.
- **Sistemas Off-Grid e On-Grid:** Podem ser integradas a sistemas conectados à rede elétrica ou independentes, permitindo o armazenamento da energia gerada ou a sua utilização direta na rede.
- **Eletrificação rural e remota:** Em áreas rurais ou remotas onde a infraestrutura elétrica convencional é limitada, as placas solares oferecem uma fonte de energia acessível e sustentável.
- **Veículos e dispositivos móveis:** São utilizadas em veículos solares, como barcos, carros e aeronaves, além de dispositivos móveis e carregadores portáteis movidos a energia solar.

AGIR

Desafio:

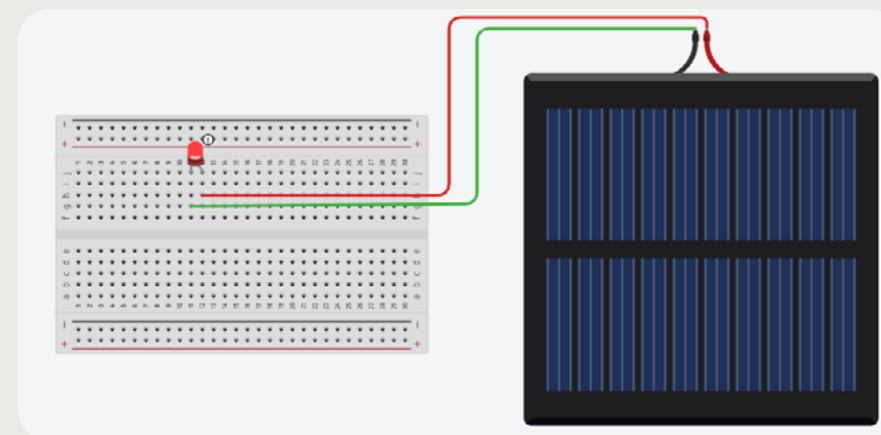
Objetivo: Esse desafio consiste em demonstrar na prática a aplicabilidade das placas solares.

Sugestão: Defina temas e peça para que os alunos elaborem maquetes.

Proposta de solução

Materiais necessários:

- Uma protoboard;
- Placas solares;
- Jumpers;
- LEDs;
- Materiais para montagem da maquete.

Montagem:

Fonte: Equipe Ed-mundo

Resultado:

Fonte: https://www.youtube.com/watch?app=desktop&v=F-GfpAIKWNL8